

1-0

3. To start in the middle of a file:

```
od -bcx a.out +100.
```

This displays `a.out` in octal-byte, character, and hexadecimal formats, starting from the 100th byte. The `.` (dot) after the offset makes it a decimal number. Without the dot, the dump would start from the 64th (100 octal) byte.

Related Information

The following commands: “**sdb**” on page 875 and “**pg**” on page 744.

“Overview of International Character Support” in *Managing the AIX Operating System*.

The discussion of Japanese Language Support in *Japanese Language Support User's Guide*.

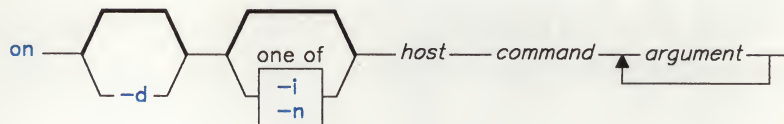
on

on

Purpose

Executes a command remotely when NFS is installed on your system.

Syntax



OL805490

Description

The **on** program executes commands on another system. All environment variables are passed. The current working directory is preserved if the working file is mounted on the host or exported to it. Relative path names work only if they are within the current file system. Because commands are issued at one machine and executed on another, absolute path names can cause problems.

The **on** command's standard input passes as standard input to the remote program, unless you use the **-n** flag. The **on** command's files receive standard output and standard error from the remote command.

Note: The **on** command cannot be used to execute commands requiring superuser authority.

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

Flags

- d** Displays status messages as work progresses.
- i** Uses remote echoing and special character processing. This flag should be used when remotely executing interactive programs, such as **vi**. Terminal modes are active when this flag is used.

- n** Does not pass standard input to the remote standard input. This causes the remote program to get an end-of-file message when it reads standard input. The **on -n** program is used to run commands in the background with job control.

File

/etc/exports

Related Information

The following command, “**rexd**” on page 832.

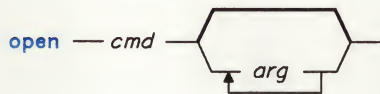
open

open

Purpose

Opens a virtual terminal.

Syntax



OL805337

Description

The **open** command opens a virtual terminal and runs the specified *file* on that terminal. If *file* does not reside in one of the directories specified in the shell **PATH** variable, you must give a full path name. Any arguments that follow *file* on the command line are passed to *file*. To move from one virtual terminal to another, press Next Window (**Alt-Action**). To close a virtual terminal, press **END OF FILE (Ctrl-D)** or end the application that is running on it.

Notes:

1. Be sure to use **Alt-Action** to check for open virtual terminals and **Ctrl-D** to close them before you log off.
2. You can run the **actman** command before opening any virtual terminals to help you keep track of virtual terminals so you can close all of them before you log off.
3. If you are in a trusted shell and issue the command **opensh**, then the new virtual terminal is opened, but not activated. You must use **Alt-Action** to activate the new virtual terminal.

Usability Services Commands

The following additional commands are available to you from within the Usability Services Activity Manager (**/usr/bin/actmgr**):

- | | |
|-----------------|--|
| hide | Removes an activity window from the window ring. |
| activate | Activates an activity window. |
| cancel | Cancels an activity window. |

For details about using these commands, see *Usability Services Reference* or *Usability Services Guide*.

Example

To run another shell on a new virtual terminal:

```
open sh
```

To move back and forth between this new virtual terminal and any others that you have opened, press Next Window (**Alt-Action**). To close this terminal and log off the new shell, press END OF FILE (**Ctrl-D**).

Related Information

The following command: “**actman**” on page 32.

“Using Display Station Features” in *Using the AIX Operating System*.

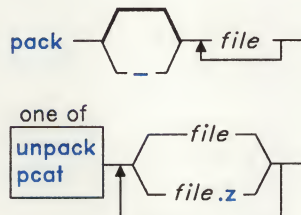
pack

pack

Purpose

Compresses files.

Syntax



OL805061

Description

pack

The **pack** command stores the specified *file* in a compressed form. The input file is replaced by a packed file with a name derived from the original file name (*file.z*), with the same access modes, access and modification dates, and owner as the original file. Directories cannot be compressed.

The input file name can contain no more than 12 bytes to allow space for the added *z* extension.

If **pack** cannot create a smaller file, it stops processing and reports that it is unable to save space. (A failure to save space, generally happens with small files or files with uniform character distribution.) The amount of space saved depends on the size of the input file and the character frequency distribution. Because a decoding tree forms the first part of each *.z* file, you will generally not be able to save space with files smaller than three blocks. Typically, text files are reduced 25 to 40 percent.

The exit value of the **pack** command is the number of files that it could not pack. Packing is not done under any one of the following conditions:

- The file is already packed.
- The input file name has more than 12 bytes.
- The file has links.
- The file is a directory.
- The file cannot be opened.

- No storage blocks are saved by packing.
- A file called *file.z* already exists.
- The *.z* file cannot be created.
- An I/O error occurs during processing.

Note: Both **pcat** and **unpack** operate only on files ending in *.z*. As a result, when you specify a file name that does not end in *.z*, **pcat** and **unpack** add the suffix and search the directory for a file name with that suffix.

pcat

The **pcat** command reads the specified *files*, unpacks them, and writes them to standard output. (Japanese Language Support does not support the **pcat** command.)

unpack

The **unpack** is the reverse of the **pack** command. It reads the input *files*, expands them, and writes them to their original file name—that is, the name without the *.z* suffix.

The exit value of **pcat** is the number of files it was unable to unpack. A file cannot be unpacked if any one of the following occurs:

- The file name (exclusive of the *.z*) has more than 12 bytes.
- The file cannot be opened.
- The file is not a packed file.

The **unpack** command expands files created by **pack**. For each file specified, **unpack** searches for a file called *file.z*. If this file is a packed file, **unpack** replaces it by its expanded version. The **unpack** command names the new file name by removing the *.z* suffix from *file*. The new file has the same access modes, access and modification dates, and owner as the original packed file.

The exit value is the number of files the **unpack** command was unable to unpack. A file cannot be unpacked if any one of the following occurs:

- The file cannot be opened.
- The file is not a packed file.
- A file with the unpacked file name already exists.
- The unpacked file cannot be created.

Note: The **unpack** command writes a warning to standard output if the file it is unpacking has links. The new unpacked file has a different i-node than the packed file from which it was created. However, any other files linked to the packed file's original i-node still exist and are still packed.

pack

Flag

- Displays statistics about the input *files*. The statistics are calculated from a Huffman minimum redundancy code tree built on a byte-by-byte basis. Repeating - (minus) on the command line toggles this function.

Examples

1. To compress files:

```
pack chap1 chap2
```

This compresses chap1 and chap2, replacing them with files named chap1.z and chap2.z. **pack** displays the percent decrease in size for each file.

2. To display statistics about the amount of compression done:

```
pack - chap1 - chap2
```

This compresses chap1 and chap2 and displays statistics about chap1, but not about chap2. The first - (minus) turns on the statistic display, and the second turns it off.

3. To display compressed files:

```
pcat chap1.z chap2 | pg
```

This displays the compressed files chap1.z and chap2.z on the screen in expanded form, a page at a time (! pg). Note that **pcat** added the .z to the end of chap2, even though we did not enter it.

4. To use a compressed file without expanding the copy stored on disk:

```
pcat chap1.z | grep 'Greece'
```

This pipes the contents of chap1.z in its expanded form to the **grep** command. See page 914 for a discussion of piping.

5. To expand compressed files:

```
unpack chap1.z chap2
```

This expands the compressed files chap1.z and chap2.z, replacing them with files named chap1 and chap2. Note that you can give **unpack** file names either with or without the .z suffix.

Related Information

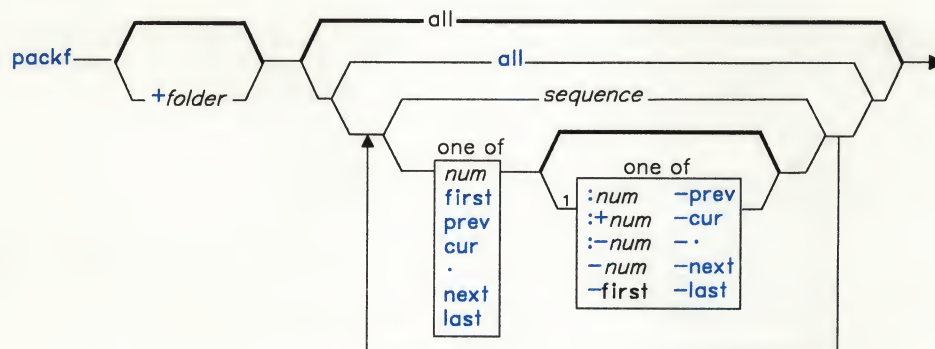
The following command: “**cat**” on page 137.

packf

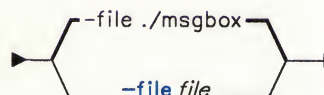
Purpose

Compresses the contents of a folder into a file.

Syntax



AJ2FL212



packf -help

AJ2FL213

¹ Do not put a blank between these items.

OL805308

Description

The **packf** command is used to move messages from a folder and compress those messages into a single file. You can unpack packed messages by using the **inc** command. **packf** is part of the MH (Message Handling) package and can be used with other MH and AIX commands.

packf

The **packf** command takes the specified messages from the specified folder and appends them to the specified file. If the file does not exist, **packf** displays a prompt that asks if you want to create the file. If you want to create the file, **packf** creates the file and places the messages in the file. **packf** separates each message with four **Ctrl-A** characters and a New line character.

Flags

- file file** Places the messages at the end of the specified file. If *file* does not exist, **packf** asks you if you want to create the file. If you want to create the file, **packf** creates *file*. The default file is *./msgbox*.
- +folder msgs** Specifies the messages that you want to pack. *msgs* can be several messages, a range of messages, or a single message. You can use the following message references when specifying *msgs*:
- | | | |
|-------------|--------------|-----------------|
| <i>num</i> | first | prev |
| cur | . | next |
| last | all | <i>sequence</i> |
- The default is all of the messages in the current folder. If several messages are specified, the first message packed becomes the current message.
- help** Displays help information for the command.

Profile Entries

- Current-Folder:** Sets your default current folder.
Msg-Protect: Sets the protection level for your new message files.
Path: Specifies your *user-mh-directory*.

Files

- \$HOME/.mh-profile* The MH user profile.

Related Information

The MH command “**inc**” on page 518.

The **mh-profile** file in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

passwd

Purpose

Changes login password.

Syntax

`passwd — user —`

OL805206

Description

The **passwd** command establishes or changes the password for your login user name. When you enter this command, you get a prompt for the old password if one exists, and two successive prompts for the new password. You must enter the same password twice for it to take effect. Passwords can be up to eight ASCII characters long. Only the password owner or the superuser can change a password. To change a password, the owner must know the old password. Passwords are subject to restrictions established by the person who administers the system.

Note: The password files must be on the same node.

Files

<code>/etc/passwd</code>	Password file; contains user IDs.
<code>/etc/opasswd</code>	Previous version of the password file.
<code>/etc/security/passwd</code>	Password field; contains encrypted passwords.
<code>/etc/security/opasswd</code>	Previous version of the security password file.
<code>/etc/security/config</code>	Defines password restrictions.

Related Information

The following commands: “**login**” on page 584 and “**users, adduser**” on page 1129.

The **getpwent** subroutine and the **passwd** file in *AIX Operating System Technical Reference*.

The discussion of password security in *Managing the AIX Operating System*.

The discussion of Japanese Language Support in *Japanese Language Support User's Guide*.

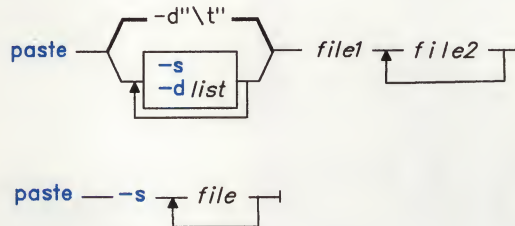
paste

paste

Purpose

Merges the lines of several files or subsequent lines in one file.

Syntax



OL805207

OL805366

Description

The **paste** command reads input *files* (standard input if you specify a - as a file name), concatenates the corresponding lines of the given input files, and writes the resulting lines to standard output. Output lines are restricted to 511 characters.

Without a flag, or with the **-d** flag, **paste** treats each file as a column and joins them horizontally with a tab character by default (parallel merging). You can think of **paste** as the counterpart of the **cat** command (see page 137), which concatenates files vertically, that is, one file after another.

With the **-s** flag, **paste** combines subsequent lines of an input file (serial merging). These lines are joined with the tab character by default.

Note: The action of **pr -t -m** is similar to that of **paste**, but creates extra blanks, tabs and lines for a nice page layout.

Japanese Language Support Information

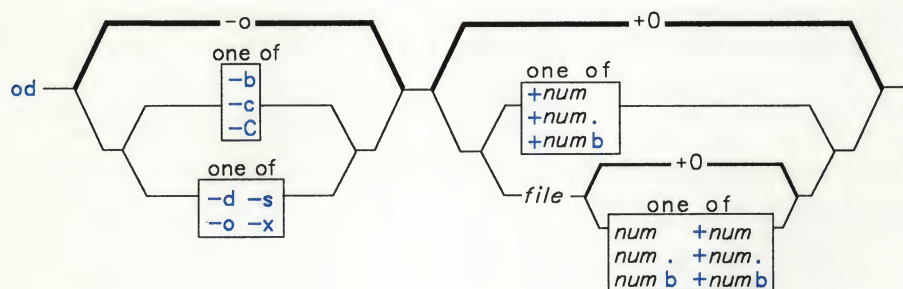
This command has not been modified to support Japanese characters.

od

Purpose

Writes the contents of storage to the standard output.

Syntax



OL805205

¹ Do not put a blank between these items.

OL805308

Description

The **od** command reads *file* (standard input by default), and it writes to standard output the information stored in *file* using the format specified by the first flag. If you do not specify the first flag, **-o** is the default.

When **od** reads standard input, *num* must be preceded by a + (plus sign).

Flags

- b Displays bytes as octal values.
- c Displays bytes as ASCII characters. The following nongraphic characters appear as C escape sequences:
 - \0 Null
 - \b Backspace
 - \f Form feed
 - \n New-line character

\r Return

\t Tab

\s1

\s2

\s3

\s4 Extended character shifts. (When Japanese Language Support is installed on your system, extended character shifts are not supported.)

Others appear as three-digit octal numbers.

- C Displays any extended characters as standard printable ASCII characters, using the appropriate character escape string.
-

Japanese Language Support Information

- C Displays any SJIS characters in hexadecimal form.
-

-d Displays 16-bit words as unsigned decimal values.

-o Displays 16-bit words as octal values.

-s Displays 16-bit words as signed decimal values.

-x Displays 16-bit words as hexadecimal values.

The *num* parameter specifies the point in the file where the storage output starts. The *num* parameter is interpreted as octal bytes. If a . (dot) is added to *num*, it is interpreted in decimal. If *b* is added to *num*, it is interpreted in blocks of 512 bytes.

The storage output continues until the end of the file.

Examples

1. To display a file in octal a page at a time:

```
od a.out | pg
```

This displays **a.out** in octal (base 8) word format a page at a time.

2. To translate a file into several formats at once:

```
od -cx a.out >a.xcd
```

This writes **a.out** in hexadecimal (base 16) format (-x) into the file **a.xcd**, giving also the ASCII character equivalent, if any, of each byte (-c).

Flags

- dlist** Changes the delimiter that separates corresponding lines in the output with one or more characters in *list* (the default is a tab). If more than one character is in *list*, then they are repeated in order until the end of the output. In parallel merging, the lines from the last file always end with a new-line character, instead of one from *list*.

The following special characters can also be used in *list*:

<code>\n</code>	New-line character
<code>\t</code>	Tab
<code>\\</code>	Backslash
<code>\0</code>	Empty string (not a null character).
<code>c</code>	An extended character.

You must quote characters that have special meaning to the shell.

- s** Merges subsequent lines from the first file horizontally. With this flag, **paste** works through one entire file before starting on the next. When it finishes merging the lines in one file, it forces a new line and then merges the lines in the next input file, continuing in the same way through the remaining input files, one at a time. A tab separates the lines unless you use the **-d** flag. Regardless of the *list*, the last character of the file is forced to be a new-line character.

Examples

- To paste several columns of data together:

```
paste names places dates > npd
```

This creates a file named `npd` that contains the data from `names` in one column, `places` in another, and `dates` in a third. If `names`, `places`, and `dates` look like:

names	places	dates
rachel	New York	February 5
jerry	Austin	March 13
mark	Chicago	June 21
marsha	Boca Raton	July 16
scott	Seattle	November 4

then `npd` contains:

paste

```
rachel New York February 5
jerry Austin March 13
mark Chicago June 21
marsha Boca Raton July 16
scott Seattle November 4
```

A tab character separates the name, place, and date on each line. As in this example, the columns do not always line up because the tab stops are set at every eighth column.

2. To separate the columns with a character other than a tab:

```
paste -d"!@" names places dates > npd
```

This alternates ! and @ as the column separators. If names, places, and dates are the same as in Example 1, then npd contains:

```
rachel!New York@February 5
jerry!Austin@March 13
mark!Chicago@June 21
marsha!Boca Raton@July 16
scott!Seattle@November 4
```

3. To display the standard input in multiple columns:

```
ls | paste - - - -
```

This lists the current directory in four columns. Each - tells **paste** to create a column containing data read from the standard input. The first line is put in the first column, the second line in the second column, . . . , the fifth line in the first column, and so on.

This is equivalent to:

```
ls | paste -d"\t\t\t\n" -s -
```

which fills the columns across the page with subsequent lines from the standard input. The -d"\t\t\t\n" defines the character to insert after each column: a tab character (\t) after the first three columns, and a new-line character (\n) after the fourth. Without the -d flag, **paste -s -** would display all of the input as one line with a tab between each column.

Related Information

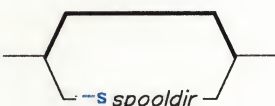
The following commands: “**grep**” on page 501, “**cut**” on page 269, and “**pr**” on page 761. “Overview of International Character Support” in *Managing the AIX Operating System*.

pcnfs

Purpose

Serves PC-NFS client requests.

Syntax

`/etc/rpc.pcnfsd` 

OL805499

Description

The **pcnfsd** daemon handles PC-NFS client requests for services, such as authentication and print spooling, on remote machines. PC-NFS is a program that allows machines running the Personal Computer Disk Operating System (PC-DOS) to be networked with machines running NFS. See *Managing the AIX Operating System* for information on PC-NFS.

The **pcnfsd** daemon starts when the **inetd** server starts if the appropriate entry is placed in the **inetd.conf** file. Print spooling requests are then sent to the default print default print spooling directory, which is usually requests is If you plan to use a directory other than the default print spooling directory (**/usr/tmp**), you cannot start **pcnfsd** from **inetd.conf**. Instead, you must specify the directory (using the **-2** flag) and start **pcnfsd** from the command line.

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

Flags

-s spooldir Specifies the name of the directory designated to receive print spooling requests for PC-NFS clients. When this flag is used, the **pcnfsd** command must be issued from the command line. If the **spooldir** name is not specified, the default print spooling directory is used.

pcnfsd

File

/etc/inetd.conf TCP/IP configuration file.

Related Information

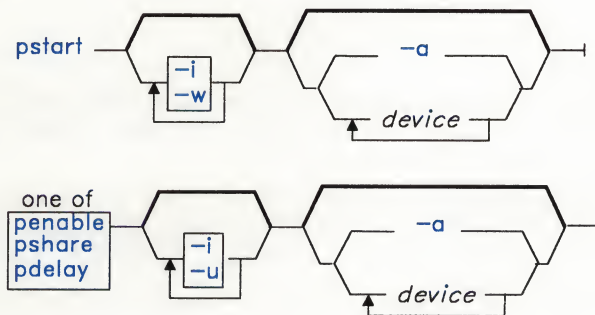
The section on configuring **pcnfsd** in *Managing the AIX Operating System*.

pdisable, phold

Purpose

Disables or reports the availability of login ports.

Syntax



OL805208

Description

The **pdisable** and **phold** commands each disable a set of login ports. No one can log in on a disabled port. The system disables a port by updating an entry in the `/etc/portstatus` file and then sending a signal to **init**. When **init** receives the signal and reads the updated status entry, it takes the appropriate action.

Use the *device* parameter to specify the ports to be enabled. Permitted values for *device* include:

- A full device name, such as `/dev/tty1`.
- A simple device name, such as `tty1`.
- A general class of devices in the form *attribute=value*, which is equivalent to naming each port with a stanza in `/etc/ports` that includes the specified attribute).

If you do not specify a *device* to disable, each command report the names of currently disabled ports in its set.

pdisable

pdisable

The **pdisable** command disables the specified port. Even if a user is logged in at that port, subsequent log in is not permitted. To disable the port, the system ends **logger**. If you do not specify any arguments, **pdisable** reports the names of all disabled ports.

phold

The **phold** command allows logged-in users to continue, but does not allow any more users to log in. If you do not specify any arguments, **phold** reports the names of all ports on hold.

Flags

- a With **pdisable**, disables all ports that are currently enabled in the **/etc/portstatus** file. With **phold**, holds all ports that are currently enabled in the **/etc/portstatus** file.
- i Reinitializes an existing **/etc/portstatus** file instead of updating the existing one. You typically use this flag in the **/etc/rc** command file to re-establish default port enabling before starting up the system with multiple users.
- w Makes the command return immediately rather than wait for **init** to confirm the changes in port status.

Examples

1. To display the names of all ports currently disabled:
`pdisable`
2. To disable all ports that are enabled in **/etc/portstatus**, even if users are logged in:
`pdisable -a`
3. To disable the work station attached to the **/dev/tty8** port:
`pdisable tty8`
4. To disable the work station attached to the **/dev/tty2** and make the command return immediately rather than wait for **init** to confirm port status:
`pdisable -w /dev/tty2`
5. To list the ports that are currently on hold:
`phold`
6. To put all 9600 baud ports on hold:
`phold speed=9600`

Files

/etc/locks	Contains lock files for pshare and pdelay .
/etc/ports	Contains descriptions of known normal, shared, and delayed login ports.
/etc/portstatus	Contains current status of each known login port.

Related Information

The following commands: “**init**” on page 521 and “**pstart, penable, pshare, pdelay**” on page 791.

The **ports** and **portstatus** files in *AIX Operating System Technical Reference*.

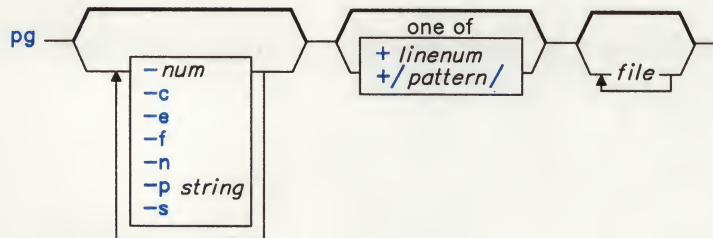
pg

pg

Purpose

Formats files to the work station.

Syntax



OL805245

Description

The **pg** command reads *files* and writes them to standard output one screen at a time. If you specify *file* as **-** (minus) or run **pg** without arguments, **pg** reads standard input. Each screen is followed by a prompt. If you press the **Enter** key, another page is displayed. The **pg** command lets you back up to review something that has already passed.

To determine work station attributes, **pg** scans the file **terminfo** for the work station type specified by the environment variable **TERM**. The default type is **dumb**. See *AIX Operating System Technical Reference* for information on **terminfo**.

Subcommands

When **pg** pauses and issues its prompt, you can issue a subcommand. Some of these subcommands change the display to a particular place in the file, some search for specific patterns in the text, and others change the environment in which **pg** works.

The following commands display a selected place in the file:

<i>page</i>	Displays the specified <i>page</i> .
+ num	Displays the page <i>num</i> pages after the current page.
- num	Displays the page <i>num</i> pages before the current page.
l	Scrolls the display one line forward.

<i>numl</i>	Displays a screen with the specified line <i>number</i> at the top.
<i>+numl</i>	Scrolls the display <i>num</i> lines forward.
<i>-numl</i>	Scrolls the display <i>num</i> lines backward.
d	Scrolls half a screen forward. Pressing Ctrl-D also does this.
-d	Scrolls half a screen backward. Pressing -Ctrl-D also does this.
Ctrl-L	Displays the current page again. A single period also does this.
\$	Displays the last page in the <i>file</i> . Do not use this when the input is from a pipeline.

The following commands search for text patterns in the text. You can use the patterns described in “**ed**” on page 371. They must always end with a new-line character, even if the **-n** flag is used. In an expression such as **[a-z]**, the minus means “through” according to the current collating sequence. A collating sequence may define *equivalence classes* for use in character ranges. See “Overview of International Character Support” in *Managing the AIX Operating System* for more information on collating sequences and equivalence classes.

[num]/pattern/ Search for the *numth* occurrence of *pattern*. The search begins immediately after the current page and continues to the end of the current file, without wraparound. The default for *num* is 1.

num?pattern?
num^pattern^ Search backward for the *numth* occurrence of *pattern*. The searching begins immediately before the current page and continues to the beginning of the current file, without wraparound. The ^ (circumflex) is useful for the Adds 100 work station, which cannot handle the ?. The default for *num* is 1.

After searching, **pg** normally displays the line found at the top of the screen. You can change this by adding **m** or **b** to the search command to leave the line found in the middle or at the bottom of the window with all succeeding subcommands. Use the suffix **t** to return to displaying the line with the pattern to the top of the screen.

You can change the **pg** environment with the following subcommands:

[num]n	Begins examining the <i>numth</i> next file in the command line. The default <i>num</i> is 1.
[num]p	Begins examining the <i>numth</i> previous file on the command line. The default <i>num</i> is 1.
[num]w	Displays another window of text. If <i>num</i> is present, sets the window size to <i>num</i> .
s file	Saves the input in <i>file</i> . Only the current file being examined is saved. This command must always end with a new-line character, even if you specify the -n flag.

- h** Displays an abbreviated summary of available subcommands.
- q or Q** Quits **pg**.
- !AIX-cmd** Sends the specified AIX command to the shell named in the **SHELL** environment variable. If this is not available, the default shell is used. This command must always end with a new-line character, even if the **-n** flag is used.

At any time when output is being sent to the work station, you can press QUIT WITH DUMP (**Ctrl-V**) or INTERRUPT (**Alt-Pause**). This causes **pg** to stop sending output and displays the prompt. Then you can enter one of the preceding commands in the normal manner.

Note: Some output is lost when when you press QUIT WITH DUMP (**Ctrl-V**) or INTERRUPT (**Alt-Pause**) because any characters waiting in the output queue are purged when the dQUIT signal is received.

If standard output is not a work station, **pg** acts like the **cat** command, except that a header is displayed before each file.

While waiting for work station input, **pg** stops running when you press INTERRUPT (**Alt-Pause**). Between prompts these signals interrupt the current task and place you in the prompt mode.

Note: If work station tabs are not set every eight positions, unpredictable results can occur.

Flags

- c** Moves the cursor to the home position and clears the screen before each page. This flag is ignored if **clear-screen** is not defined for your work station type in the **terminfo** file.
- e** Does not pause at the end of each file.
- f** Does not split lines. Normally, **pg** splits lines longer than the screen width.
- n** Stops processing when a **pg** command letter is entered. Normally, commands must end with a new-line character.
- p string** Uses *string* as the prompt. If the *string* contains a **%d**, the **%d** is replaced by the current page number in the prompt. The default prompt is **:** (colon). If *string* contains spaces, you must quote it.
- s** Highlights all messages and prompts.
- + linenum** Starts at *linenum*.
- num** Specifies the number of lines in the window. On work stations that contain 24 lines, the default is 23.
- + /pattern/** Starts at the first line that contains *pattern*.

Files

/usr/lib/terminfo/*
/tmp/pg*

Related Information

The following commands: “**ed**” on page 371 and “**grep**” on page 501.

The **terminfo** file in *AIX Operating System Technical Reference*.

“Overview of International Character Support” in *Managing the AIX Operating System*.

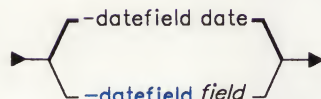
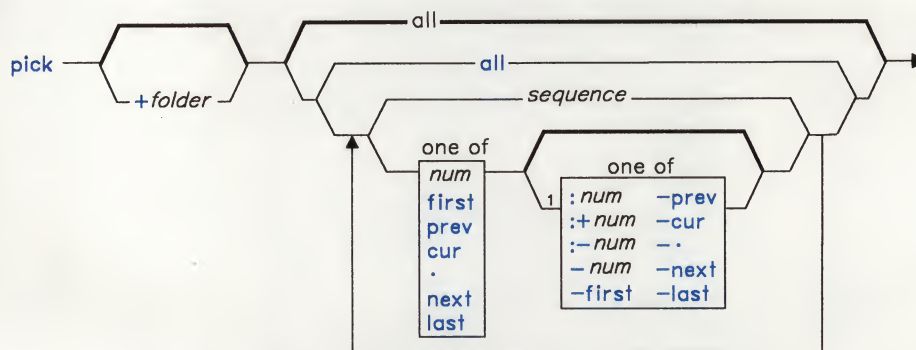
pick

pick

Purpose

Selects messages by content, and creates and modifies sequences.

Syntax

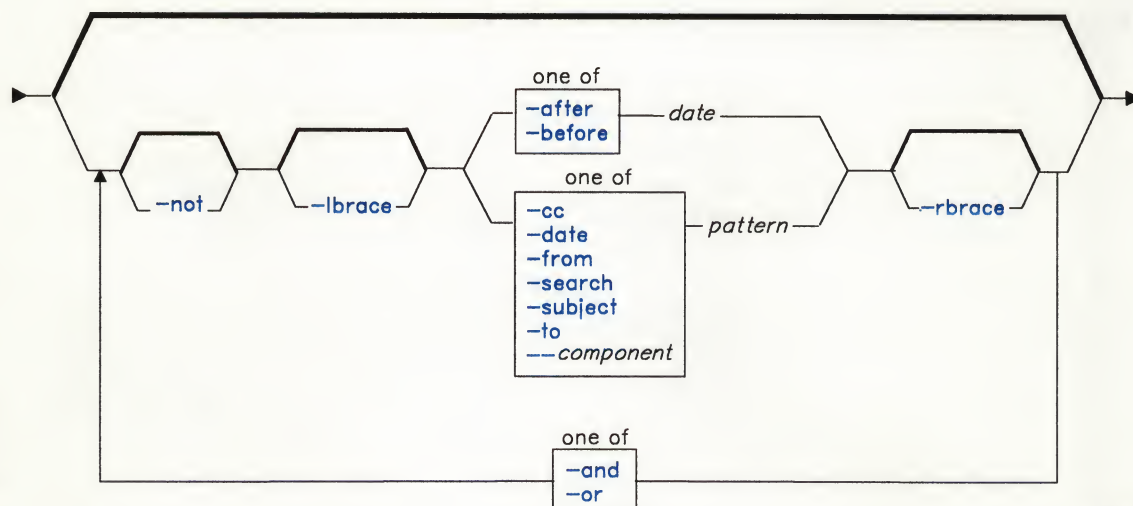


¹ Do not put a blank between these items.

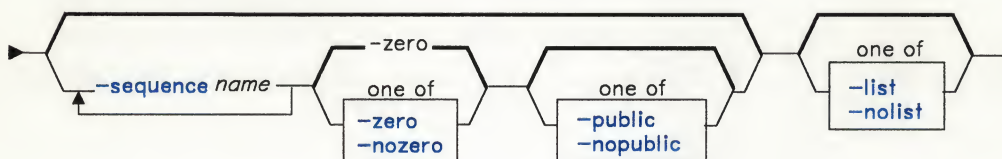
AJ2FL216

AJ2FL161

OL805308



AJ2FL217



pick — -help —|

AJ2FL253

Description

The **pick** command is used to select messages and put them into sequences. The **pick** command is part of the MH (Message Handling) package and can be used with other MH and AIX commands.

The **pick** command allows you to select messages that contain particular character patterns or that have particular dates. You can use the **-and**, **-or**, **-not**, **-lbrace**, and **-rbrace** flags to construct compound conditions for selecting messages.

pick

Flags

-after date	Selects messages with dates later than the specified date. You can use the following date specifications for <i>date</i> :		
	yesterday	today	tomorrow
	sunday	monday	tuesday
	wednesday	thursday	friday
	saturday	-dd	sysdate
	The pick command treats the days of the week as days in the past. For example, monday means last Monday, not today or next Monday. You can use the -dd argument to specify a number of days in the past. For example, -31 means 31 days ago. For sysdate you can specify any valid format defined for your system. See “ date ” on page 281 for more information about date formats.		
-and	Forms a logical AND operation between two message selecting flags (for example, pick -after Sunday -and -from mark). -and has precedence over -or , but -not has precedence over -and . Use the -lbrace and -rbrace flags to override this precedence.		
-before date	Selects messages with dates earlier than the specified date. See the -after flag for how to specify <i>date</i> .		
-cc pattern	Selects messages that contain the character string <i>pattern</i> in the Cc: field.		
-date pattern	Selects messages that contain the character string <i>pattern</i> in the Date: field.		
-datefield field	Specifies which dated field is parsed when the -after and -before flags are given. By default, pick uses the Date: field.		
+folder msgs	Specifies the messages that you want to search. <i>msgs</i> can be several messages, a range of messages, or a single message. You can use the following message references when specifying <i>msgs</i> :		
	num	first	prev
	cur	.	next
	last	all	sequence
	The default for <i>msgs</i> is all . The default folder is the current folder. If you specify a folder, that folder becomes the current folder.		
-from pattern	Selects messages that contain the character string <i>pattern</i> in the From: field.		
-help	Displays help information for the command.		

-lbrace	Groups -and , -or , and -not operations. Operations between the -lbrace and -rbrace flags are evaluated as one operation. You can nest the -lbrace and -rbrace flags.
-list	Sends a list of selected message numbers to standard output. This allows you to use the pick command to generate message numbers to use as input for other commands. For example, <code>scan 'pick -after tuesday -list'</code> scans all messages in the current folder that were sent after last Tuesday. -list is the default if no sequence is specified.
-nolist	Keeps the pick command from generating a list of the selected message numbers (see the -list flag). If a sequence is specified, -nolist is the default.
-npublic	Restricts the specified sequence to your usage. -npublic does not restrict the messages in the sequence, only the sequence. This option is the default if the folder is write-protected from other users.
-not	Forms a logical NOT operation on a message selecting flag (for example, <code>pick -not -from george</code>). This construction evaluates to all messages not chosen by the message selecting flag. -not has precedence over -and , and -and has precedence over -or . Use the -lbrace and -rbrace flags to override this precedence.
-nozero	Appends the selected messages to the specified sequence (see the -zero flag).
-or	Forms a logical OR operation on two message-selecting flags (for example, <code>pick -from amy -or -from mark</code>). -not has precedence over -and , and -and has precedence over -or . Use the -lbrace and -rbrace flags to override this precedence.
-public	Makes the specified sequence available to other users. -public does not make protected messages available, only the sequence. This option is the default if the folder is not write-protected from other users.
-rbrace	Groups -and , -or , and -not operations. Operations between the -lbrace and -rbrace flags are evaluated as one operation. You can nest the -lbrace and -rbrace flags.
-search <i>pattern</i>	Selects messages that contain the character string <i>pattern</i> anywhere in the message.
-sequence <i>name</i>	Stores the messages selected by the pick command in the sequence <i>name</i> .
-to <i>pattern</i>	Selects messages that contain the character string <i>pattern</i> in the To: field.

pick

- zero** Clears the specified sequence before placing the selected messages into the sequence. This flag is the default (see the **-nozero** flag).
- component *pattern*** Selects messages that contain the character string *pattern* in the heading field *component* (for example, pick --reply-to amy).

Profile Entries

- Current-Folder:** Sets your default current folder.
- Path:** Specifies your *user-mh-directory*.

Files

- \$HOME/.mh-profile** The MH user profile.

Related Information

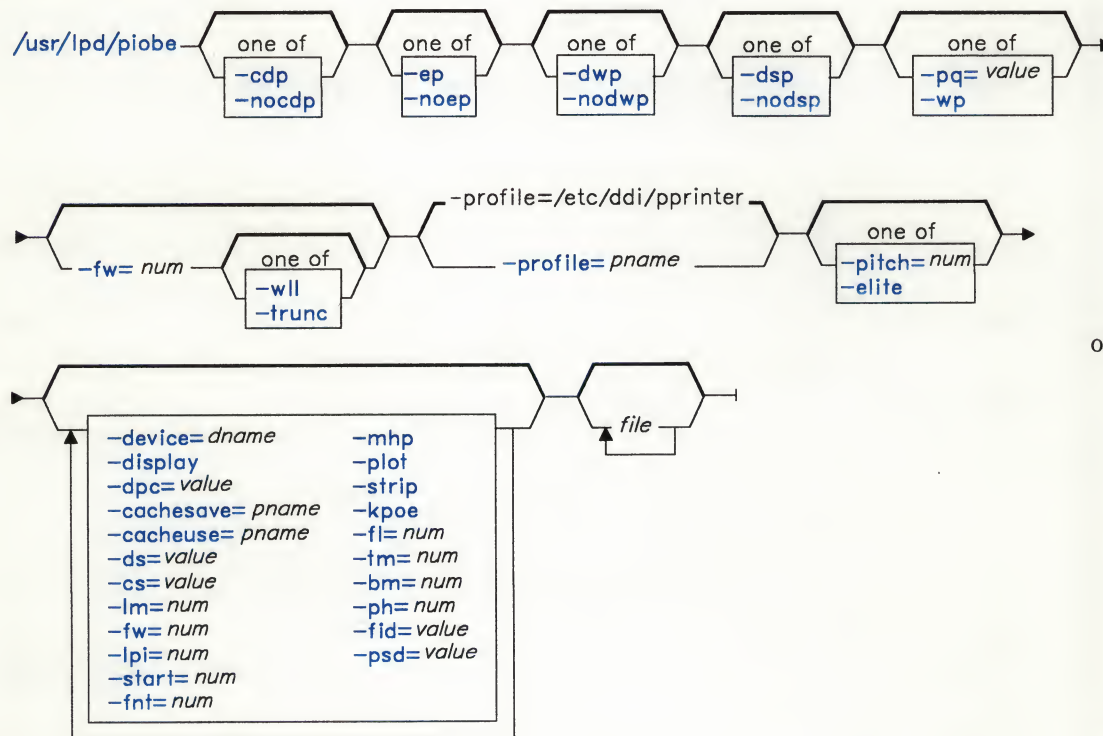
- The MH command “**mark**” on page 637.
- The **mh-profile** file in *AIX Operating System Technical Reference*.
- “Overview of the Message Handling Package” in *Managing the AIX Operating System*.

piobe

Purpose

Writes a file to standard output in a format suitable for sending to a line printer.

Syntax



OL805391

OL805452

Description

The **piobe** command writes *file* to its standard output in a form that is suitable for a line printer. If you do not specify a *file*, **piobe** reads standard input. **piobe** is normally called by the **qdaemon** command after you have enqueued a file with the **print** command (see “**print**” on page 767). The **qdaemon** directs the output from **piobe** to the appropriate device.

Flags

You can specify the following flags on the **print** command line or in the **/etc/qconfig** file (see *AIX Operating System Technical Reference*).

-bm = num	Sets the bottom margin to <i>num</i> lines from the top of the page.
-cachesave = pname	Specifies the path name of the file where the font Personal PagePrinter adapter is to be saved after the input data stream prints.
cacheuse = pname	Specifies the path <i>nname</i> of the font cache file loaded in the IBM Personal PagePrinter adapter before the input data stream is printed.
-cdp	
-nocdp	Turns the condensed printing mode on (-cdp) or off (-nocdp).
-cs = value	Uses PC code set 1 or 2 .
-device = dname	Specifies the name of a printer stanza in the printer configuration file (see "Files" on page 756).
-display	Specifies that the input data stream has KSR code page controls.
-dpc = value	Prints in the specified color. Valid color <i>values</i> are red , blue , yellow , and black .
-ds = value	Specifies how the input data stream is to be printed on the IBM 4216 Personal PagePrinter. Valid values are the following: ps ps ppxl Proprinter XL emulation any This is the default. ps is assumed if the first character of the data stream is % . Otherwise, Proprinter XL emulation is assumed.
-dsp	
-nodsp	Turns the double strike mode on (-dsp) or off (-nodsp).
-dwp	
-nodwp	Turns the double wide printing mode on (-dwp) or off (-nodwp).
-elite	Sets the character pitch to 12 , the same as specifying -pitch = 12 .
-ep	
-noep	Turns the emphasized printing mode on (-ep) or off (-noep).
-fid = value	Specifies the font identifier for an IBM 5202 Quietwriter® III Printer font. Valid values for embedded fonts are 11 (Courier 10), 85 (Courier 12), 254 (Courier 17), and 159 (Boldface).

	Values for fonts in the pluggable cartridges precede the font name on the cartridge label.
-fl = num	Sets the form length to <i>num</i> .
-fnt = num	Allows font change. Valid values for <i>num</i> are 1 through 8.
-fw = num	Sets the right margin at <i>num</i> characters from the left edge of the carriage.
-kpoe	Forgives keying mistakes and ignores invalid flags. If you specify this flag, piobe processes the job and sends you no message. If you do not specify this flag, piobe does not forgive invalid flags and does not print the job. In this case, it sends you a message detailing the error.
-lm = num	Sets the left margin at <i>num</i> characters from the left edge of the carriage.
-lpi = num	Sets the number of lines per inch to <i>num</i> . Valid settings are 6 and 8.
-mhp	Allows the horizontal position on the print line to be maintained for line feed and vertical tab controls, if desired.
-ph = num	Allows you to use single-sheet paper in the Quietwriter printer. The printer stops at the end of each page, beeps three times, and waits for you to push the start button. <i>num</i> can have the following values: <div> <div>0</div> <div>Manual operation.</div> <div>1</div> <div>Sheetfeed operation.</div> <div>2</div> <div>Continuous operation.</div> </div>
-pitch = num	Sets the character pitch to <i>num</i> .
-plot	Specifies that the input data is to be passed through without modification. This allows arbitrary files to be printed on arbitrary printers.
- = value	Specifies the printer name. Messages that report a printer is out of paper or needs attention include the name of the printer. The default name of the printer is specified in the <i>/etc/qconfig</i> file.
-pq = value	Prints in specified print quality. Valid quality <i>values</i> are dp , text , and letter .
-profile = pname	Specifies the name of a printer configuration file. The default name is <i>/etc/ddi/pprinter</i> .

piobe

-psd = <i>value</i>	Specifies a paper source drawer for the optional IBM 5202 Quietwriter III Printer two-drawer sheetfeeder. Valid values are 1 (top drawer), 2 (bottom drawer), and 3 (envelopes).
-start = <i>num</i>	Sets the starting page number to <i>num</i> .
-strip	Strips all multibyte controls from the data stream. This flag is useful in filter mode in order to send data that has imbedded printer controls to a nonprinter device.
-tm = <i>num</i>	Sets the top margin to <i>num</i> lines.
-trunc	Specifies that lines exceeding the value set by -fw should be truncated.
-wll	Specifies that lines exceeding the value set with the -fw flag should overflow to the next line. This is the reverse of the -trunc flag.
-wp	Selects word processing mode, the same as specifying -pq = letter .

Files

/etc/ddi/pprinter	Parallel configuration information.
/etc/ddi/sprinter	Serial configuration information.

Related Information

The following commands: “**print**” on page 767 and “**qdaemon**” on page 802.

The **qconfig** file in *AIX Operating System Technical Reference*.

portmap

Purpose

Maps RPC programs to the servicing ports on RPC servers

Syntax

`portmap` `—l`

A5AC5022

Description

The **portmap** program, also known as the **portmapper**, maps RPC program number to the port numbers on RPC servers in order for NFS clients to make RPC service calls. When an RPC server starts, it contacts the **portmap** program to register the RPC programs it is prepared to serve and the port on which it is listening for calls. When NFS clients call an RPC procedure for a given program number, the clients contact the **portmap** program to determine the port number to which they should send the packets.

The **portmap** program is usually started from an entry in the **inetd.conf** file.

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

Files

`/etc/portmap`
`/etc/rc.tcpip`

Related Information

The following command: “**rpcinfo**” on page 845.

The section on NFS in *Managing the AIX Operating System*.

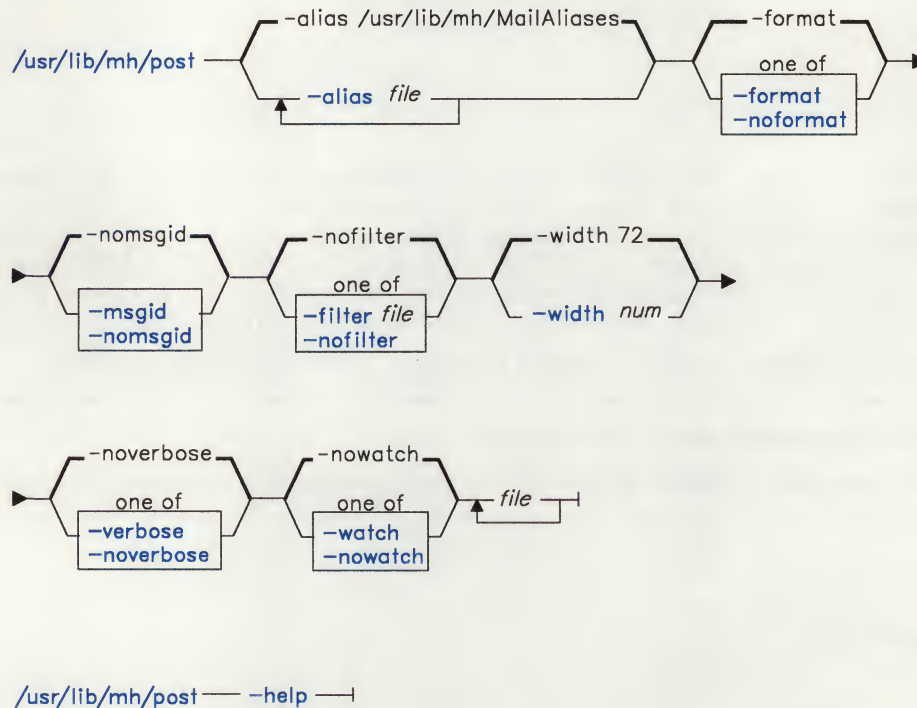
post

post

Purpose

Delivers a message.

Syntax



AJ2FL233

Description

The **post** command is used to route messages to the proper destinations. **post** is not designed to be run directly by the user; it is designed to be called by other programs. The **post** command is part of the MH (Message Handling) package.

The **post** command searches all components of a message that specify a recipient's address and parses each address to check for proper format. **post** puts proper addresses in the

standard format and calls the **sendmail** command. **post** also performs header operations, such as appending the **Date:** and **From:** components and processing the **Bcc:** component.

The **post** command may report errors when parsing complex addresses (for example, @A:harold@B.UUCP). If the **sendmail** program is installed on your system and you use complex addresses, use the **spost** command instead of the **post** command.

Flags

- alias file** Searches the specified mail alias file for addresses. You can use this flag repetitively to specify multiple mail alias files. **post** automatically searches the file **/usr/lib/mh/MailAliases**.
- filter file** Uses the header components in the specified file for copies of the message sent to **Bcc:** recipients.
- format** Puts all recipient addresses in a standard format for the delivery transport system. This flag is the default.
- help** Displays help information for the command.
- msgid** Adds a message-identification component (such as **Message-ID:**) to the message.
- nofilter** Strips the **Bcc:** header from the message for the **To:** and **cc:** recipients. Sends the message with minimal headers to the **Bcc:** recipients. This flag is the default.
- noformat** Does not alter the format of the recipient addresses.
- nomsgid** Does not add a message-identification component to the message. This flag is the default.
- noverbose** Does not display information during the delivery of the message to the **sendmail** command. This flag is the default.
- nowatch** Does not display information during delivery by the **sendmail** command. This flag is the default.
- verbose** Displays information during the delivery of the message to the **sendmail** command. This information allows you to monitor the steps involved.
- watch** Displays information during the delivery of the message by the **sendmail** command. This information allows you to monitor the steps involved.
- width num** Sets the width of components that contain addresses. The default is 72 columns.

post

Files

/usr/lib/mh/MailAliases
/usr/lib/mh/mtstailor

The default mail alias file.
The MH tailor file.

Related Information

The following commands: “**ali**” on page 48, “**conflict**” on page 196, “**mhmail**” on page 646, “**send**” on page 893, “**sendmail**” on page 897, “**spost**” on page 978, and “**whom**” on page 1222.

The **mh-alias**, **mh-format**, **mh-mail**, and **mh-profile** files in *AIX Operating System Technical Reference*.

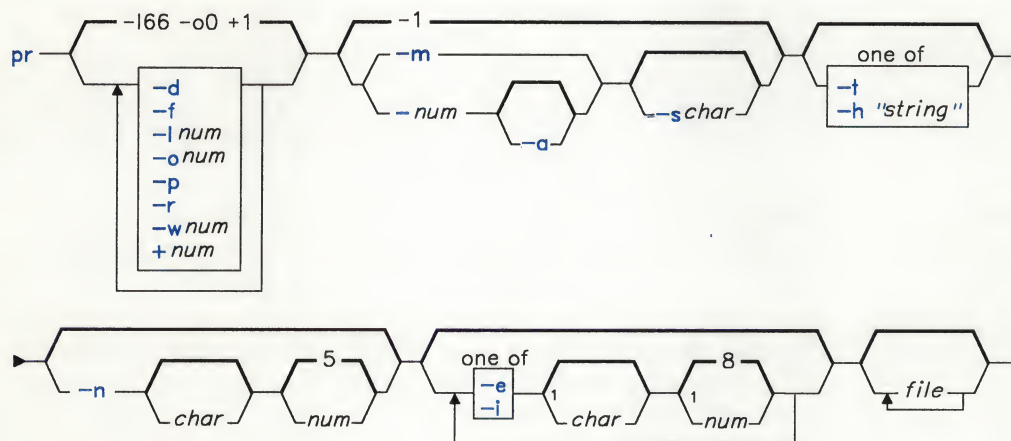
“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

pr

Purpose

Writes a file to standard output.

Syntax



¹ Do not put a blank between these items.

OL805437

Description

The **pr** command writes *file* to the standard output. If you do not specify *file* or if *file* is a `-` (minus), **pr** reads standard input. A heading that contains the page number, date, time, and the name of the file separates the output into pages.

Unless specified, columns are of equal width and separated by at least one space. Lines that are too long for the page width are cut off. If the standard output is a work station, **pr** does not display any error messages until it has ended.

Flags

- a** Displays multicolumn output across the page.
- d** Double-spaces the output.

- e[*char*][*num*]** Expands tabs to character positions $num + 1$, $2 * num + 1$, $3 * num + 1$, and so on. The default value of *num* is 8. Tab characters in the input expand to the appropriate number of spaces to line up with the next tab setting. If you specify *char* (any character other than a digit) that character becomes the input tab character. The default value of *char* is the ASCII TAB character.
- f** Uses a form feed character to advance to a new page. (Otherwise **pr** issues a sequence of line feed characters.) Pauses before beginning the first page if the standard output is a work station.
- h "*string*"** Displays *string* as the page header instead of the file name. The flag and string should be separated by a blank.
- i[*char*][*num*]** In the *output*, replaces white space wherever possible by inserting tabs to character positions $num + 1$, $2 * num + 1$, $3 * num + 1$, and so on. The default value of *num* is 8. If you specify *char* (any character other than a digit), that character becomes the output tab character. (The default value of *char* is the ASCII TAB character).
- l*num*** Sets the length of a page to *num* lines (the default is 66).
- m** Combines and writes all files at the same time, with each file in a separate column. (This overrides the *-num* and *-a* flags).
- n[*char*][*num*]** Provides *num*-digit line numbering (the default value of *num* is 5). The number occupies the first $num + 1$ character positions of each column of normal output or each line of *-m* output. If you specify *char* (any character other than a digit), that character is added to the line number to separate it from whatever follows (the default value of *char* is an ASCII TAB character).
- o*num*** Indents each line by *num* character positions (the default is 0). The number of character positions per line is the sum of the width and offset.
- p** Pauses before beginning each page if the output is directed to a work station. (**pr** sounds the alarm at the work station and waits for you to press the **Enter** key.)
- r** Does not display diagnostic messages if the system cannot open files.
- s*char*** Separates columns by the single character *char* instead of by the appropriate number of spaces (the default for *char* is an ASCII TAB character).
- t** Does not display the five-line identifying header and the five-line footer. Stops after the last line of each file without spacing to the end of the page.
- w*num*** Sets the width of a line to *num* character positions (the default value is 72 for equal-width multicolumn output, no limit otherwise).

- `-num` Produce *num*-column output (the default is 1). The `-e` and `-i` flags are assumed for multicolumn output.
- `+num` Begin the display with page *num* (the default value is 1).

Examples

1. To print a file with headings and page numbers on the printer:

```
pr prog.c | print
```

This adds page headings to `prog.c` and sends it to the **print** command. The heading consists of the date the file was last modified, the file name, and the page number.

2. To specify a title:

```
pr -h "MAIN PROGRAM" prog.c | print
```

This prints `prog.c` with the title `MAIN PROGRAM` in place of the file name. The modification date and page number are still printed.

3. To print a file in multiple columns:

```
pr -3 word.lst | print
```

This prints the file `word.lst` in three vertical columns.

4. To print several files side by side on the paper:

```
pr -m -h "Members and Visitors" member.lst visitor.lst | print
```

This prints `member.lst` and `visitor.lst` side by side with the title `Members and Visitors`.

5. To modify a file for later use:

```
pr -t -e prog.c > prog.notab.c
```

This replaces tab characters in `prog.c` with blanks and puts the result in `prog.notab.c`. Tab positions are at columns 9, 17, 25, 33, The `-e` tells **pr** to replace the tab characters; the `-t` suppresses the page headings.

pr

Files

/dev/tty* To suspend messages.

Related Information

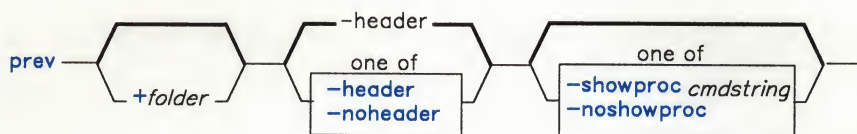
The following command: “**cat**” on page 137.

prev

Purpose

Shows the previous message.

Syntax



prev — -help —|

AJ2FL162

Description

The **prev** command is used to display the previous message in a folder. **prev** is equivalent to the **show** command with **prev** specified as the message. **prev** is part of the MH (Message Handling) package and can be used with other MH and AIX commands.

The **prev** command links to the **show** program and passes **show** its flags and attributes.

Note: If you link to **prev** and call that link something other than **prev**, your link will function like the **show** command, rather than like the **prev** command.

The **show** command invokes a program to perform the listing. The system default is **/bin/pg**. You can define your own default with the **showproc:** entry in **\$HOME/.mh-profile**. If you set **showproc:** entry to **mhl**, **show** calls an internal **mhl** routine instead of the **mhl** command. You can also specify the program to perform a listing in the *cmdstring* of the **-showproc** flag.

The **show** command passes any flags that it does not recognize to the program performing the listing. Thus, you can specify flags for the listing program, as well as the flags described in this command section.

Flags

+folder	Specifies the folder that contains the message you want to show.
-header	Displays a one-line description of the message being shown. The description includes the folder name and the message number.
-help	Displays help information for the command.
-noheader	Does not display a one-line description of each message being shown.
-noshowproc	Uses /bin/cat to perform the listing.
-showproc cmdstring	Uses the specified command string to perform the listing.

Profile Entries

Current-Folder:	Sets your default current folder.
Path:	Specifies your <i>user_mh_directory</i> .
showproc:	Specifies the program used to show messages.

Files

<code>\$HOME/.mh-profile</code>	The MH user profile.
---------------------------------	----------------------

Related Information

Other MH commands: “**next**” on page 694, “**show**” on page 942.

The **mh-format**, **mh-mail**, and **mh-profile** files in *AIX Operating System Technical Reference*.

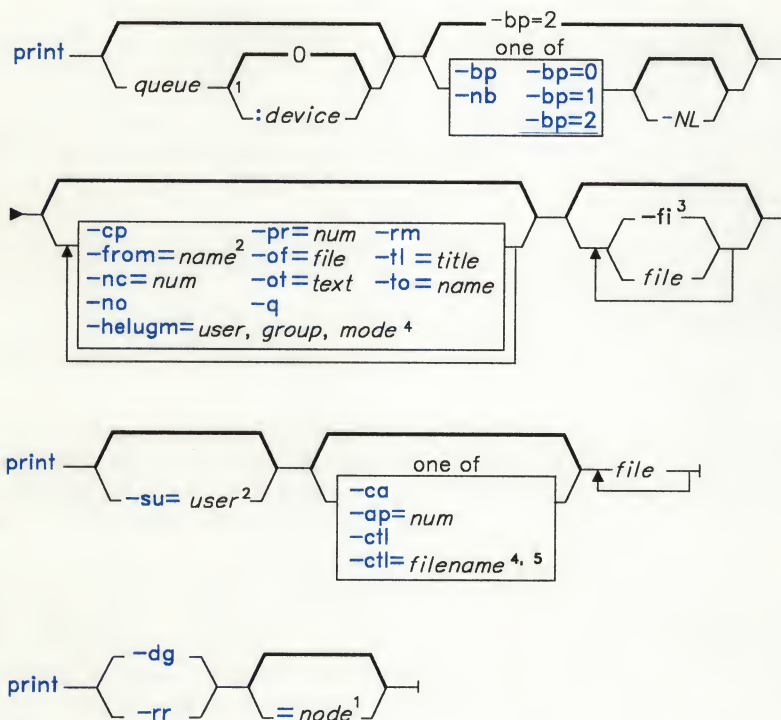
“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

print

Purpose

Enqueues a file.

Syntax

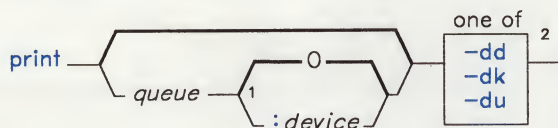


OL805348

OL805147

OL805354

print



¹ Do not put a blank between these items.

² Only members of the system group can use these flags.

³ Only specify `-fi` once.

OL805328

Description

The **print** command is a general purpose utility for enqueueing requests to a shared resource, typically a printer device. With the **print** command you can do the following:

- Enqueue print requests
- Cancel print requests
- Alter the priority of print requests
- Display the status of print queues and devices
- Specify the printing of security labels on hard copy.

For complete information about security labels, see the discussion of printer security labels in *Managing the AIX Operating System*.

To enqueue files on a specific queue, specify `-queue`. If more than one device services a queue, you can also request a particular device by specifying `:device` after the name of the queue. If you do not specify a device, the job is sent to the first available device. If you do not specify a *file*, **print** copies standard input into a file and enqueues it for printing.

Print requests can have operator messages associated with them. This feature is useful in a distributed services environment. You can use messages to tell the user operating the printer to load a special form into the printer before allowing this job to print. These messages are specified with the `-of` and `-ot` flags. When **qdaemon**, the daemon that processes print requests, is ready to begin a request that has an associated message, the system displays the message on the console of the machine where **qdaemon** is running. The text of the message is accompanied by a prompt that tells the printer operator how to signal the request to continue or cancel the request.

Notes:

1. Before you can print a file, you must have read access to it. To remove a file, you must also have write access to the directory that contains the file.
2. If you want to continue changing the *file* after you issue the **print** command but before it is printed, you must use the **-cp** flag.
3. The operator must respond to messages sent with the **-of** and **-ot** flags. Print requests with these flags stop processing until the operator responds.
4. When enqueueing files on a printer, flags and file names can be interspersed in any order.
5. Blanks between flags and their arguments are not permitted.

Flags

If you give **print** a list of *file* names, it enqueues them all for printing on the default printer.

-ap = num

Changes to *num* the priority of the named file. The file must have been submitted for printing prior to entering the **print** command with this flag. See “-pr” on page 770 for a description of priorities.

-bp = num

-bp

-nb

Controls the printing of burst pages according to the value of *num* as follows:

- 0** Does not print headers or trailers.
- 1** Prints one header page before each *file*. No trailer appears.
- 2** Prints a header page at the beginning and a trailer page at the end of each *file*.

The **header** stanza in the **qconfig** file defines the default treatment of burst pages.

Specifying only **-bp** is the same as specifying **-bp = 2**. Specifying **-nb** is the same as specifying **-bp = 0**.

-ca

Cancels the printing of the named *files*.

-ctl[= filename]

Specifies security labels and access codes for a file located on a remote system. You must be a superuser to use this flag.

-cp

Copies the file. Ordinarily, to save disk space, **print** remembers the name of the file, but does not actually copy the file itself. Use the **-cp** flag if you want to continue changing the file while you are waiting for the current copy to be printed.

print

-fi	Causes print to act as a filter. The print command automatically reads standard input if you do not specify <i>files</i> as arguments. However, if you specify <i>file</i> arguments, you can use the -fi flag to force print to read standard input at the appropriate time.
-nc = num	Prints <i>num</i> copies of the file. Normally a file prints only once.
-nl	Suppresses top-of-page and bottom-of-page output labeling.
-no	Notifies you when your job is finished. If the -to flag is also used, print notifies the user for whom the request is intended (see the -to flag on page 770).
-of = file	Submits an operator message with a print request. The specified <i>file</i> contains the text of the message.
-ot = text	Submits an operator message with a print request. The specified <i>text</i> contains the text of the message.
-pr = num	Sets the priority of the named <i>file</i> to <i>num</i> . Higher numbers assign higher priority. The default priority is 15. The maximum priority is 20 for most users and 30 for the users with superuser authority and members of the system group (group 0).
-q	Displays the status of the queues and printers. You can specify an <i>argname</i> to view a single queue. The environment variable NLTIME controls the appearance of the time field.
-rm	Removes the file after it prints.
-tl = title	Puts <i>title</i> on the header page and displays it when the -q flag is specified. Normally the job title is the name of the file. If print reads from standard input, the job title is STDIN.# where # is the process ID of the print command.
-to = name	Labels the output for delivery to <i>name</i> . Normally the output is labeled for delivery to the person issuing the print request.

In addition to the previous flags that are available to all users, the **print** command accepts the following flags when they are entered by users who have superuser authority or users who are members of the system group:

-dd	Turns off the device associated with <i>queue</i> . The qdaemon no longer sends jobs to the device, and entering print -q shows its status as OFF. Any job currently running on the device is allowed to finish.
-dg[= node]	Kills the qdaemon after all currently running jobs are finished. Use of this flag is the only clean way to bring the qdaemon down. Use of the kill command may cause problems, such as jobs hanging up in the queue. If the qdaemon is on a remote node, specify the node. You can specify it as either a node ID or nickname. If no node is specified, the local node is assumed.

- dk** Acts the same as **-dd**, except current jobs are killed. They remain in the queue, and are run again when the device is turned on.
- du** Turns on the device associated with *queue*. The **qdaemon** sends jobs to it again and entering **print -q** shows its status as READY.

Note: If more than one device is associated with a queue, you must specify the device as well as the queue when you use the **-dd**, **-dk**, and **-du** flags. Devices are numbered, starting at zero, in the order that they appear in the **qconfig** file. For example, **-lp:0** designates the first device on the **lp** queue. **-lp** designates the same device only if there is no other device on that queue.
- from = name** Labels the output as though *name* had submitted it. You can only use this flag with superuser authority.
- rr[= node]** Forces the **qdaemon** to reread the **qconfig** file after all currently running jobs are finished. With this flag, a user with superuser authority can change the **qconfig** file without having to kill and restart the **qdaemon**. If the **qdaemon** is on a remote node, specify the node either a node ID or nickname. If no node is specified, the local node is assumed.
- su = user** Cancels or changes the priority on another *user*'s job when used with the **-ca** or the **-ap** flags. For example, a job report submitted by user *ann* can be cancelled as follows:


```
print -su=ann -ca report
```

The **print** command passes flags it does not recognize to the backend that does the printing. Thus, for each queue there are flags not described above that can be included on the **print** command line. See "**piobe**" on page 753 for a list of these flags.

Examples

1. To print a file on the default printer:

```
print memo
```

2. To print a file with page numbers:

```
pr prog.c | print
```

The **pr** command puts a heading at the top of each page that includes the date the file was last modified, the name of the file, and the page number. The **print** command then prints the file.

3. To see if a file is still waiting to be printed:

```
print -q
```


print

This command displays the status of the queues and printers. If a file has not yet printed, then it appears in the queue status listing. If you piped data to **print**, as in Example 2, then it is listed as PRIMARY.OUTPUT.

4. To stop printing a file:

```
print -ca chapter1
```

This command cancels the request you made earlier to print the file `chapter1`. If the file is currently being printed, then the printer stops immediately. If the file has not yet printed, then it is removed from the queue so that it will not be printed. If the file is not in the queue, **print** displays the message:

```
no such request from you -- perhaps it's done?
```

5. To disconnect a printer from the queueing system:

```
print -a:2 -dd
```

This command stops print requests from being sent to the third printer that serves the `-a` queue. If a file is currently being printed, it is allowed to finish. You must be a member of the **system** group (group 0) to run this command.

Note: The printers serving a given queue are numbered starting with zero in the order that they appear in the `/etc/qconfig` file.

Files

<code>/etc/qdaemon</code>	Queueing daemon.
<code>/usr/lpd/qdir/*</code>	Queue requests.
<code>/usr/lpd/stat/*</code>	Information on the status of the devices.
<code>/usr/spool/qdaemon/*</code>	Temporary copies of enqueued files.
<code>/etc/qconfig</code>	Queue configuration file.
<code>/etc/security/config</code>	Specifies output labelling information.

Related Information

The following commands: “**piobe**” on page 753, “**pr**” on page 761, and “**qdaemon**” on page 802.

The **qconfig** file in *AIX Operating System Technical Reference*.

The discussion of the printer subsystem in *Managing the AIX Operating System*.

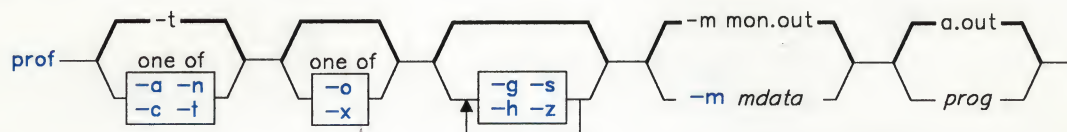
“Overview of International Character Support” in *Managing the AIX Operating System*.

prof

Purpose

Displays program profile data.

Syntax



OL805438

Description

The **prof** command interprets profile data collected by the **monitor** subroutine for the object file **prog** (**a.out** by default). It reads the symbol table in the object file *prog* and correlates it with the profile file (**mon.out** by default). **prof** displays, for each external text symbol, the percentage of execution time spent between the address of that symbol and the address of the next, the number of times that function was called, and the average number of milliseconds per call.

For the number of calls to a function to be tallied, you must have compiled the file using the **-p** flag of the **cc** command. This flag also arranges for the object file to include a special profiling startup function that calls the **monitor** subroutine at the beginning and end of execution. It is the call to **monitor** at the end of execution that writes **mon.out**. Thus, only programs that explicitly **exit** or **return** from **main** cause the **mon.out** file to be produced.

Note: No more than 600 functions can have call counters established during program execution. If you exceed this limit, **prof** overwrites other data and damages the **mon.out** file. **prof** automatically reports the number of call counters used whenever the number exceeds 500.

Japanese Language Support Information

This command has not been modified to support Japanese characters.

prof

Flags

The mutually exclusive flags **a**, **c**, **n**, and **t** determine how **prof** sorts the output lines:

- a** Sorts by increasing symbol address.
- c** Sorts by decreasing number of calls.
- n** Sorts lexically by symbol name.
- t** Sorts by decreasing percentage of total time (default).

The mutually exclusive flags **o** and **x** specify how to display the address of each symbol monitored.

- o** Displays each address in octal, along with the symbol name.
- x** Displays each address in hexadecimal, along with the symbol name.

Use the following flags in any combination:

- g** Includes nonglobal symbols (static functions). This option requires object code that was compiled with the **-g** flag.
- h** Suppresses the heading normally displayed on the report. (This is useful if the report is to be processed further.)
- m** *mdata* Takes profiling data from *mdata* instead of **mon.out**.
- s** Displays a summary of monitoring parameters and statistics on standard error.
- z** Includes all symbols in the profile range, even if associated with zero calls and zero time.

Files

mon.out	Default profile.
a.out	Default object file.

Related Information

The following commands: “**cc**” on page 140 and “**nm**” on page 705.

The **exit** and **profil** system calls and the **monitor** subroutine in *AIX Operating System Technical Reference*.

profiler

Purpose

Profiles the operating system.

Syntax

prfld — */unix* — *kernel-image* —

prfstat — *on* — *off* —

prfdc — *file* — *minutes* — *hour* —

prfsnap — *file* —

prfpr — *file* — *cutoff* — */unix* — *kernel-image* —

OL805006

Description

With the **prfld**, **prfstat**, **prfdc**, **prfsnap**, and **prfpr** commands, you can examine the activity of the AIX Operating System.

profiler

prfld

Use **prfld** to initialize the recording mechanism in the system. It produces a table containing the starting address of each system subroutine as extracted from *kernel-image*.

Japanese Language Support Information

This command has not been modified to support Japanese characters.

prfstat

Use **prfstat** to enable or disable the sampling mechanism. **prfstat** also reveals the number of text addresses being measured.

Japanese Language Support Information

This command has not been modified to support Japanese characters.

prfdc, prfsnap

Use **prfdc** and **prfsnap** to collect profiler data by copying the current value of all the text address counters to a file where the data can be analyzed. **prfdc** stores the counters into *file* every specified *minutes* and turns off at *hour* (0-24). **prfsnap** collects data at the time of invocation only, adding the counter values to *file*.

Japanese Language Support Information

This command has not been modified to support Japanese characters.

prfpr

Use **prfpr** to format the data collected by **prfdc** or **prfsnap**. It converts each text address to the nearest text symbol (as found in *kernel-image*) and displays it if the percent activity for that range is greater than *cutoff*.

Japanese Language Support Information

This command has not been modified to support Japanese characters.

Files

/dev/prf	Interface to profile data and text addresses.
/unix	System kernel image file.

Related Information

The **prf** file in *AIX Operating System Technical Reference*.

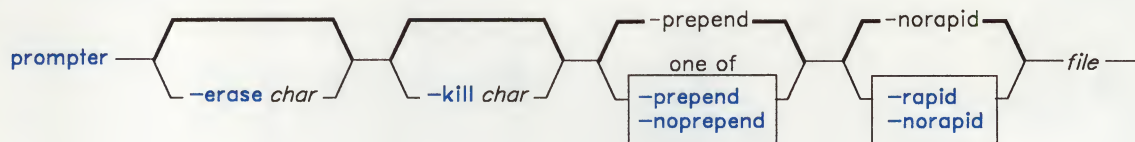
prompter

prompter

Purpose

Invokes a prompting editor.

Syntax



`prompter` — `-help` — |

AJ2FL250

Description

The **prompter** command is used to invoke the prompter editor for message entry. **prompter** is not designed to be run directly by the user; it is designed to be called by other programs. The **prompter** command is part of the MH (Message Handling) package.

The **prompter** command opens the specified file and scans it for empty components such as **To:** and prompts you to fill in those fields. If you press **Enter** without adding text, **prompter** later deletes the component.

After the first blank line or line of dashes in the file, **prompter** accepts text for the body of the message. If the body already contains text and the flag **-noperpend** is specified, **prompter** displays the text followed by the message:

-----Enter additional text

prompter appends any text that is entered to the message body. If you specify the **-prepend** flag, **prompter** displays the following message instead:

-----Enter initial text

When you press **END OF FILE**, **prompter** ends text entry and returns control to the calling program.

Flags

-erase <i>char</i>	Sets the character to be used as the erase character. You can specify the octal representation of the character in the form <code>\nnn</code> , or you can specify the character itself.
-help	Displays help information for the command.
-kill <i>char</i>	Sets the character to be used as the kill character. You can specify the octal representation of the character in the form <code>\nnn</code> , or you can specify the character itself.
-noprepend	Places additional text below any text already in the message body.
-norapid	Displays any text already in the message body. This is the default.
-prepend	Places additional text before any text already in the message body. This is the default.
-rapid	Does not display text already in the message body.

Profile Entries

Msg-Protect:	Sets the protection level for your new message files.
prompter-next:	Specifies the editor used after exiting prompter .

Files

<code>\$HOME/.mh-profile</code>	The MH user profile.
<code>/tmp/prompter*</code>	A temporary copy of a message.

Related Information

Other MH commands: “**comp**” on page 185, “**dist**” on page 336, “**forw**” on page 438, “**repl**” on page 821, “**whatnow**” on page 1215.

The **mh-profile** file in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

proto

proto

Purpose

Constructs a prototype file for a file system.

Syntax

`/etc/proto` — *directory* — *prefix*

OL805007

Description

The **proto** command makes a prototype file for a file system or part of a file system. Use the prototype file as input to the **mkfs** command to construct a file system according to a predefined template. The prototype file consists of a recursive directory listing of every file on the file system, with its owner, group, and protection. It also contains the file from which the prototype file is to be initialized, formatted as described in the **mkfs** command.

Specify the base directory from which the prototype file is made with *directory*. The prototype file includes the complete subtree below *directory* that is contained on the same file system as *directory*.

The *prefix* parameter is added to the names of all the initialization files, forcing the initialization files to be taken from a place other than the prototype. Before the output from **proto** can be used with **mkfs**, **mkfs** needs a start up program, a file system size, and an i-list size. Link information is not preserved with the **proto** command.

The collating sequence is determined by the **ct_collate** array in the **NLctab** subroutine.

Japanese Language Support Information

This command has not been modified to support Japanese characters.

Related Information

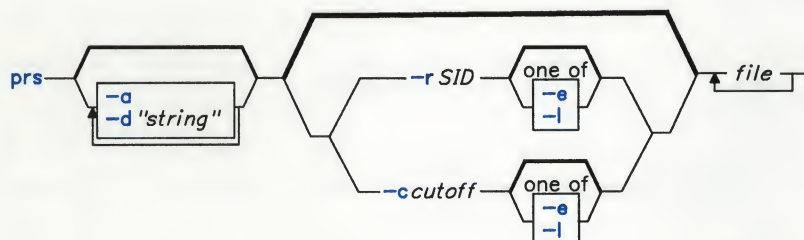
The following command: “**mkfs**” on page 658.

“Overview of International Character Support” in *Managing the AIX Operating System*.

prs

Purpose

Displays a Source Code Control System (SCCS) file.

Syntax


OL805248

Description

The **prs** command reads *files*, and writes to standard output a part or all of a **Source Code Control System (SCCS)** file. If you specify a directory in place of *file*, **prs** performs the requested actions on all SCCS files (those with a name that has the **s.** prefix). If you specify a - (minus) in place of *file*, **prs** reads standard input and interprets each line as the name of an SCCS file. **prs** continues to take input until it reads an end-of-file character.

Data Keywords

Data keywords specify which parts of an SCCS file are to be retrieved and written to standard output. All parts of an SCCS file have an associated data keyword. There is no limit to the number of times a data keyword can appear in a *string*. The information that **prs** displays consists of user-supplied text and appropriate values (extracted from the SCCS file) substituted for the recognized data keywords in the order of appearance in *string*. The format of a data keyword value is either simple, in which the keyword substitution is direct, or multiline, in which the substitution is followed by a carriage return. Text is any characters other than recognized data keywords. Specify a tab character with `\t` and a carriage return/new-line character with a `\n`. Remember to quote the `\t` and `\n` with an extra `\` to prevent shell from interpreting the `\` and just passing a `t` or `n` to **prs** as text.

The following table lists the keywords associated with information in the delta table in the SCCS file (see the **sccsfile** file in *AIX Operating System Technical Reference* for the structure of an SCCS file).

Keyword	Data Represented	Value	Format
:R:	Release number	num	Simple
:L:	Level number	num	Simple
:B:	Branch number	num	Simple
:S:	Sequence number	num	Simple
:I:	SCCS ID string (SID)	:R::L::B::S:	Simple
:Dy:	Year delta created	YY	Simple
:Dm:	Month delta created	MM	Simple
:Dd:	Day delta created	DD	Simple
:D:	Date delta created	YY/MM/DD	Simple
:Th:	Hour delta created	HH	Simple
:Tm:	Minute delta created	MM	Simple
:Ts:	Second delta created	SS	Simple
:T:	Time delta created	HH:MM:SS	Simple
:DT:	Delta type	D or R	Simple
:P:	User who created the delta	login name	Simple
:DS:	Delta sequence number	num	Simple
:DP:	Previous delta sequence number	num	Simple
:Dt:	Delta information	:DT::I::D: :T::P::DS::DP:	Simple
:Dn:	Sequence numbers of deltas included	:DS: . . .	Simple
:Dx:	Sequence numbers of deltas excluded	:DS: . . .	Simple
:Dg:	Sequence numbers of deltas ignored	:DS: . . .	Simple
:DI:	Sequence numbers of deltas included, excluded, and ignored	:Dn:/:Dx:/:Dg:	Simple
:Li:	Lines inserted by Delta	num	Simple
:Ld:	Lines deleted by Delta	num	Simple
:Lu:	Lines unchanged by Delta	num	Simple
:DL:	Delta line statistics	:Li:/:Ld:/:Lu:	Simple

Figure 7. Delta Table Keywords

Keyword	Data Represented	Value	Format
:MR:	MR numbers for delta	text	Multiline
:C:	Comments for delta	text	Multiline

Figure 7 (Part 2 of 2). Delta Table Keywords

The following table lists the keywords associated with the header flags in the SCCS file. For more information of Header flags, see Figure 1 on page 44.

Keyword	Data Represented	Value	Format
:Y:	Module type	text	simple
:MF:	MR validation flag set	yes or no	Simple
:MP:	MR validation program name	text	Simple
:KF:	Keyword/error warning flag set	yes or no	Simple
:BF:	Branch flag set	yes or no	Simple
:J:	Joint edit flag set	yes or no	Simple
:LK:	Locked releases	:R: . . .	Simple
:Q:	User defined keyword	text	Simple
:M:	Module name	text	Simple
:FB:	Floor boundary	:R:	Simple
:CB:	Ceiling boundary	:R:	Simple
:Ds:	Default SID	:I:	Simple
:ND:	Null Delta flag set	yes or no	Simple
:FL:	Header flag list	text	Multiline

Figure 8. Header Flag Keywords

The following table lists the keywords associated with other parts of the SCCS file.

Keyword	Data Represented	Value	Format
:UN:	User names	text	Multiline
:FD:	Descriptive text	text	Multiline
:BD:	Body of text	text	Multiline
:GB:	Text in a g-file	text	Multiline
:W:	A what string	:Z::M: \tab :I:	Simple
:A:	A what string	:Z::Y::M::I::Z:	Simple
:Z:	A what string delimiter	@ (#)	Simple
:F:	SCCS file name	text	Simple
:PN:	SCCS file path name	text	Simple

Figure 9. Other Keywords

Flags

Each flag or group of flags applies independently to each named file.

- a Writes information for the specified deltas, whether or not they have been removed (see “**rmDEL**” on page 837). If you do not specify the **-a** flag, **prs** supplies information only for the specified deltas that have not been removed.
- ccutoff Specifies a *cutoff* date and time for the **-e** and **-l** flags. Specify *cutoff* in the following form:
YY[MM[DD[HH[MM[SS]]]]]
All omitted items default to their maximum values, so specifying -c8402 is the same as specifying -c840229235959. You can separate the fields with any non-numeric characters. For example, you can specify -c84/2/20,9:22:25 or -c"84/2/20 9:22:25" or "-c84/2/20 9:22:25".
- d"string" Specifies the data items to be displayed. *string* is a string consisting of optional text and SCCS file data keywords. You must enclose all text and spaces in *string* in quotation marks.
- e Requests information for all deltas created *earlier* than and including the delta specified by the **-r** flag.

- l Requests information for all deltas created *later* than and including the delta specified by the -r flag.
- rSID Specifies the *SID* of a delta for which **prs** will retrieve information. If no SID is specified, **prs** retrieves the information for the SID of the highest numbered delta.

Files

/tmp/pr????

Related Information

The following commands: “**admin**” on page 41, “**delta**” on page 310, “**get**” on page 477, and “**help**” on page 513.

The **scsfile** file in *AIX Operating System Technical Reference*.

The discussion of SCCS in *AIX Operating System Programming Tools and Interfaces*.

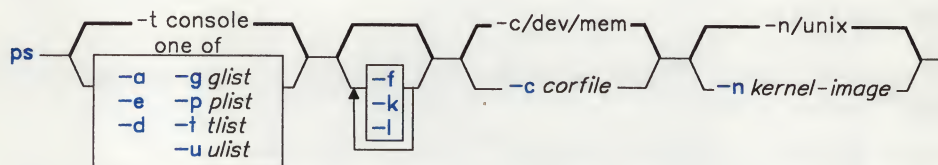
ps

ps

Purpose

Reports process status.

Syntax



OL805439

Description

The **ps** command writes certain information about active processes to standard output. Without flags, **ps** displays information about the current work station.

The column headings in a **ps** listing have the following meaning. The letters **f** and **l** following the column heads indicate which flags cause the corresponding heading to appear. If **all** follows the column head, that heading always appears. Note that the **-f** and **-l** flags determine only what information is provided about a process; they do not determine which processes are listed.

F (l)

Flags (octal and additive) associated with the process:

- 01 In core
- 02 System process
- 04 Locked in core (for example, for physical I/O);
- 10 Waiting for a page default, or forking
- 20 Being traced by another process
- 40 Another tracing flag
- 100 Process has shared text.

S (l)

The state of the process:

- 0 Nonexistent
- S Sleeping
- W Waiting
- R Running

I Intermediate
Z Canceled
T Stopped
K Available kernel process
X Growing.

UID (f,l)

The user ID of the process owner; the login name is displayed with the -f flag.

PID (all)

The process ID of the process.

PPID (f,l)

The process ID of the parent process.

C (f,l)

Processor utilization for scheduling.

STIME (f)

Starting time of the process. The **NLLDATE** and **NLTIME** environment variables control the appearance of this field.

PRI (l)

The priority of the process; higher numbers mean lower priority.

NI (l)

Nice value; used in calculating priority.

ADDR (l)

The segment number of the process stack, if normal; if a kernel process, the address of the preprocess data area.

SZ (l)

The size in blocks of the core image of the process.

WCHAN (l)

The event for which the process is waiting or sleeping; if blank, the process is running.

TTY (all)

The controlling work station for the process.

TIME (all)

The total execution time for the process.

CMD (all)

The command name; the full command name and its parameters are displayed with the -f flag.

A process that has exited and has a parent, but has not yet been waited for by the parent, is marked **<defunct>**.

With the **-f** flag, **ps** determines what the command name and parameters were when the process was created by examining memory or the paging area. If it cannot find this information, the command name, as it would appear without the **-f** flag, displays in square brackets.

Notes:

1. The process can change while **ps** is running.
2. Some data displayed for defunct processes are irrelevant.
3. The current work station is defined as the one associated with standard error. Thus redirecting standard error, for example:

```
ps 2> /dev/null
```

does not work as expected.

Flags

- | | |
|-------------------------------|--|
| -a | Writes to standard output information about all processes except the process group leaders and processes not associated with a terminal. |
| -c <i>corefile</i> | Uses <i>corefile</i> instead of the default /dev/mem . <i>corefile</i> is a core image file that has been created by the Ctrl-(left)Alt-End key sequence. |
| -d | Writes information to standard output about all processes except the process group leaders. |
| -e | Writes information to standard output about all processes except kernel processes. |
| -f | Generates a full listing. The meaning of columns in a full listing is described on page 786. |
| -g <i>glist</i> | Writes information to standard output only about processes that are in the process groups listed in <i>glist</i> . The <i>glist</i> is either a comma-separated list of process-group identifiers or a list of process-group identifiers enclosed in double quotation marks (" ") and separated from one another by a comma and/or one or more spaces. |
| -k | Writes information to standard output about kernel processes. Otherwise, it does not list kernel processes. |
| -l | Generates a long listing. The meaning of a long listing is described on page 786. |
| -n <i>kernel-image</i> | Takes <i>kernel-image</i> as the name of an alternate <i>kernel-image</i> file (/unix is the default). |

- | | |
|------------------------|--|
| -p <i>plist</i> | Displays only information about processes with the process numbers specified in <i>plist</i> . <i>plist</i> is either a comma-separated list of process-ID numbers or a list of process-ID numbers enclosed in double quotation marks (" ") and separated from one another by a comma and/or one or more spaces. |
| -t <i>tlist</i> | Displays only information about processes associated with the work stations listed in <i>tlist</i> . <i>tlist</i> is either a list of comma-separated work-station identifiers or a list of work-station identifiers enclosed in double quotation marks (" ") and separated from one another by a comma and/or one or more spaces. |
| -u <i>ulist</i> | Displays only information about processes with the user ID numbers or login names specified in <i>ulist</i> . <i>ulist</i> is either a comma-separated list of user ID's or a list of user ID's enclosed in double quotation marks (" ") and separated from one another by a comma and/or one or more spaces. In the listing, ps displays the numerical user ID unless the -f flag is used; then it displays the login name. |

Examples

1. To list the processes that you have started:

```
ps
```

This command displays a summary of information about the processes associated with your work station.

2. To display all process information available:

```
ps -e -f -l
```

This command displays all of the information (-l -f) about all processes (-e).

3. To list processes owned by specific users:

```
ps -f -l -ujim,jane,su
```

This command displays all the information available (-l -f) about the processes being run by the users jim, jane, and su.

4. To list processes associated with specific work stations:

```
ps -t-,console
```

This command displays information about processes not connected to any work station (-t-), and processes associated with the work station /dev/console.

Files

/unix	System kernel image.
/dev/mem	Memory.
/etc/passwd	Supplies UID information.
/etc/ps_data	Internal data structure.
/dev	Searched to find work station (tty) names.

Related Information

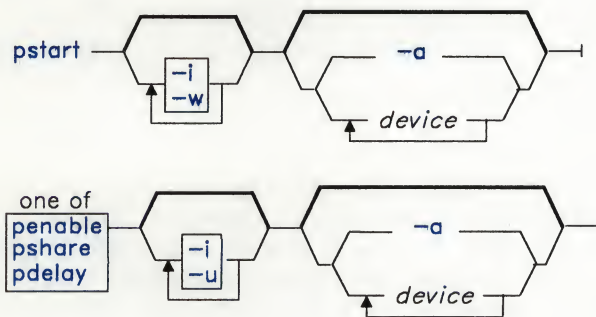
The following commands: “**kill**” on page 552 and “**nice**” on page 699.

pstart, penable, pshare, pdelay

Purpose

Enables or reports the availability of login ports.

Syntax



OL805208

Description

The **pstart**, **penable**, **pshare**, and **pdelay** commands each enable a set of login ports in the **/etc/ports** file. Enabling a port makes the port available to log in. The system enables a port by updating an entry in the **/etc/portstatus** file and then sending a signal to **init**. When **init** receives the signal and reads the updated status entry, it takes the appropriate action.

Use the *device* parameter to specify the ports to be enabled. Permitted values for *device* include:

- A full device name, such as **/dev/tty1**.
- A simple device name, such as **tty1**.
- A general class of devices in the form *attribute=value*, which is equivalent to naming each port with a stanza in **/etc/ports** that includes the specified attribute).

If you do not specify a *device* to enable, each command reports the names of currently enabled ports in its set.

pstart

The **pstart** command enables all ports (normal, shared, and delayed) that are enabled in the **/etc/ports** file. If you do not specify a *device* to enable, **pstart** reports the names of all enabled ports and tells whether they are currently enabled as normal, shared, or delayed. Usually the command is run in the form **pstart -a -i -w** from **/etc/rc** to enable all ports on a multiuser system.

penable

The **penable** command enables normal ports that are enabled in the **/etc/ports** file. Normal ports are ports that are asynchronous and only allow users to login to those ports. No outgoing use of the port is allowed while it is enabled. This command is equivalent to the statement **penable enabled = true**. If you do not specify a *device*, **penable** reports the names of the currently enabled normal ports.

pshare

The **pshare** command enables shared ports that are enabled in the **/etc/ports** file. Shared ports are bidirectional. This command is equivalent to the statement **pshare enabled = share**. If you do not specify a *device*, **pshare** reports the names of the currently enabled shared ports. To enable shared ports, **getty** attempts to create a lock file in **/etc/locks** which contains the ASCII process ID of the **getty** process. If the port is already in use by some other process, **getty** waits until the port is available and tries again.

pdelay

The **pdelay** command enables delayed ports that are enabled in the **/etc/ports** file. Delayed ports are ports that are enabled like shared ports except that the login herald is not displayed until the user types one or more characters (usually carriage returns). If the port is directly connected to a remote system or connected to an intelligent modem, the port is usually enabled as a delayed port to prevent the **getty** from talking to a **getty** on the remote side or to the modem on a local connection, thereby consuming system resources. This statement is equivalent to **pdelay enabled = delay**. If you do not specify a *device*, **pdelay** reports the names of the currently enabled delayed ports.

Flags

- a With **pstart**, this flag enables all ports enabled in the **/etc/ports** file (normal, shared, and delayed ports). With **penable**, this flag enables all normal ports that are enabled in the **/etc/ports** file. With **pshare**, this flag enables all shared ports that are enabled in the **/etc/ports** file. With **pdelay**, this flag enables all delayed ports that are enabled in the **/etc/ports** file.
- i Reinitializes an existing **/etc/portstatus** file instead of updating the existing one. You typically use this flag in the **/etc/rc** command file to re-establish default port enabling before starting up the system with multiple users.

- w Makes the command return immediately rather than wait for **init** to confirm the changes in port status. You must use this flag when running **pstart**, **penable**, **pshare**, or **pdelay** either in maintenance mode or from **/etc/rc** because **init** does not initiate loggers until the system is in normal mode.

Examples

1. To display the names of all ports (normal, shared, and delayed) currently enabled and how they are enabled:
`pstart`
2. To enable all normal, shared, and delayed ports that are enabled in **/etc/ports**, reinitialize the existing **/etc/ports**, and make the command return immediately rather than wait for **init** to confirm port status:
`pstart -a -i -w`
3. To enable the work station attached to the **/dev/tty2** port as a shared port:
`pshare /dev/tty2`
4. To display the names of the delayed ports that are currently enabled:
`pdelay`
5. To enable the work station attached to the **/dev/tty8** port as a shared port and return immediately without waiting to confirm the changes in the port status:
`pshare -w tty8`
6. To enable all 9600 baud ports as delayed:
`pdelay speed=9600`

Files

/etc/locks	Contains lock files for pshare and pdelay .
/etc/ports	Contains descriptions of known normal, shared, and delayed ports.
/etc/portstatus	Contains current status of each known login port.

Related Information

The following commands: “**init**” on page 521 and “**pdisable**, **phold**” on page 741.
The **ports** and **portstatus** files in *AIX Operating System Technical Reference*.

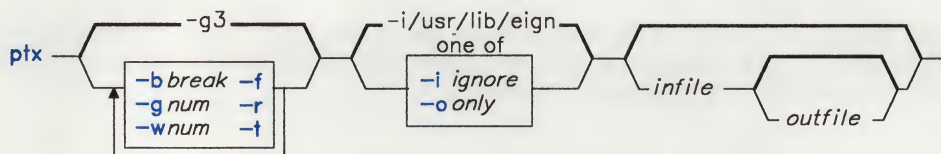
ptx

ptx

Purpose

Generates a permuted index.

Syntax



OL805250

Description

The **ptx** command reads *infile* (standard input by default), creates a permuted index from its input, and writes to *outfile* (standard output by default).

The **ptx** command searches *infile* for keywords, sorts the lines, and generates the file *outfile*. *outfile* can then be processed with **nroff** or **troff** to produce a permuted index from the file *infile*.

The **ptx** command follows three steps:

1. In the permutation, generates one line for each keyword in an input line, and rotates the keyword to the front.
2. Sorts the permuted file.
3. Rotates the sorted lines so that the keyword comes at the middle of each line.

The resulting lines in output are in the form:

.xx, "tail" "before keyword" "keyword and after" "head"

where *.xx* is an **nroff** or **troff** macro provided by the user, or provided by the **mptx** macro package (see the *AIX Operating System Technical Reference* for information on this macro package). The *before keyword* and *keyword and after* fields incorporate as much of the line as will fit around the keyword when it is printed. *tail* or *head*, at least one of which is always the empty string, are wrapped-around pieces small enough to fit in the unused space at the opposite end of the line.

Notes:

1. Line length counts do not account for overstriking or proportional spacing.
2. Lines that contain tildes (~) do not work because **ptx** uses that character internally.

Flags

- b *break* Uses the characters in the *break* file to separate words. Tab characters, new-line characters, and spaces are *always* used as break characters.
- f Does not distinguish between uppercase and lowercase characters while sorting (see “**sort**” on page 958).
- g *num* Uses *num* as the number of spaces displayed between the four parts of the line. The default *num* is 3.
- i *ignore* Does not use any words in the *ignore* file as keywords. If the -i and -o flags are not used, **/usr/lib/eign** is the default *ignore* file.
- o *only* Uses only the words in the *only* file as keywords.
- r Takes any leading nonblank characters of each input line to be a reference identifier separate from the text of the line. Attaches that identifier as a fifth field on each output line.
- t Prepares the output for the phototypesetter.
- w *num* Uses *num* as the length of the output line. The default line length is 72 characters for **nroff** and 100 for **troff**.

Files

/bin/sort
/usr/lib/eign
/usr/lib/tmac/tmac.ptx

Related Information

The following commands: “**nroff**, **troff**” on page 709 and “**troff**” on page 710.

The **mm** and **mptx** miscellaneous facilities in *AIX Operating System Technical Reference*.
“Overview of International Character Support” in *Managing the AIX Operating System*.

puttext

puttext

Purpose

Updates an output file that contains message/insert/help descriptions.

Syntax

`puttext -n infile outfile`

OL805209

Description

The **puttext** command uses the message/insert/help descriptions in *infile* to change, delete and add message/insert/help text to *outfile* for a component. (For information about the format and contents of *infile*, see *AIX Operating System Programming Tools and Interfaces*.)

The *infile* parameter specifies the name of the file where the message/insert/help descriptions reside. See *AIX Operating System Programming Tools and Interfaces* for a discussion of the **gettext** output file parameters that describes the format and contents of this file.

The *outfile* parameter specifies the name of the output file. If you specify an *outfile* that does not exist, a new component file is created. If you specify an existing *outfile*, a copy of that file is renamed as a backup file. In this case, an old backup file will be deleted.

Note: In order for the new file to be accessed by the message support run-time services, the output file name must be in the format *xxxxcc-EN.m*. If you do not specify *outfile*, the component ID is prefixed to *-EN.m* to form the output file name.

Flag

- n** Causes **puttext** to assign available index numbers to the input descriptions. If you specify this flag, all the index number fields of the input file must be underscore characters or blanks.

Related Information

The following commands: **gettext** on page 488.

The discussion of **puttext** in *AIX Operating System Programming Tools and Interfaces*.

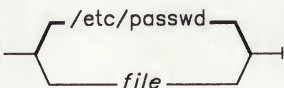
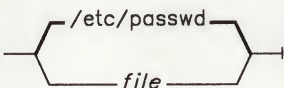
pwck

pwck

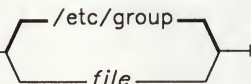
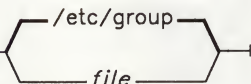
Purpose

Checks the password and group files for inconsistencies.

Syntax

pwck 
 

OL805008

grpck 
 

OL805558

Description

The **pwck** command scans the named *file* or the default file **/etc/passwd** and writes to standard output any inconsistencies. The checks include validation of the number of fields, login name, user ID, group ID, and existence of the login directory and optional program name.

The **grpck** command scans the named *file* or the default file **/etc/group** and writes to standard output any inconsistencies. The checks include validation of the number of fields, group name, group ID, and whether all login names appear in the password file. The **grpck** command writes to standard output any group entries that do not have login names.

Japanese Language Support Information

This command has not been modified to support Japanese characters.

Files

/etc/passwd	Password file; contains user IDs.
/etc/group	Group file; contains group IDs.

Related Information

The discussion of passwords in *Managing the AIX Operating System*.

pwd

pwd

Purpose

Displays the path name of the working directory.

Syntax

pwd —

OL805210

Description

The **pwd** command writes to standard output the full path name of your current directory (from the root directory). All directories are separated by a / (slash). The root directory is represented by the first /, and the last directory named is your current directory.

Related Information

The following command: “**cd**” on page 150.

The **fullstat** and **ffullstat** system calls in *AIX Operating System Technical Reference*.

pwttable

Purpose

Accesses the Distributed Services Node Security Table.

Syntax

pwttable —|

AJ2FL144

Description

The **pwttable** command lets you build, examine, or change the Distributed Services Network Node Security Table. Only members of the system group or users operating with superuser authority can use **pwttable** to change the state of the table (see “su” on page 1026). Other users can use **pwttable** to browse through the table.

Related Information

“Getting Started With Distributed Services Configuration Menus” in *Managing the AIX Operating System*.

qdaemon

qdaemon

Purpose

Schedules jobs enqueued by the **print** command.

Syntax

qdaemon —¹

¹ This command is not usually entered on the command line.

OL805148

Description

The **qdaemon** is a background process (usually started by the **rc** command file) that schedules printing jobs enqueued by **print**.

Files

/usr/lpd/qdir/*	Print requests.
/usr/lpd/stat/*	Information on the status of the devices.
/usr/spool/qdaemon/*	Temporary copies of files to be printed.

Related Information

The following commands: “**lp**” on page 593, “**piobe**” on page 753, and “**print**” on page 767.

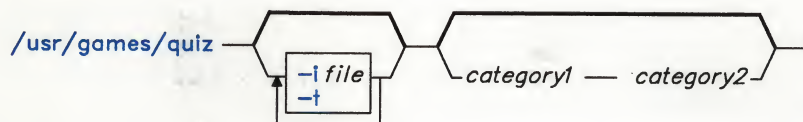
The **qconfig** file in *AIX Operating System Technical Reference*.

quiz

Purpose

Tests your knowledge.

Syntax



OL805230

Description

The **quiz** game gives associative knowledge tests on various selectable subjects. It asks about items chosen from *category1* and expects answers from *category2*. If you do not specify the categories, **quiz** gives instructions and lists the available categories.

The **quiz** game gives the correct answer whenever you press the **Enter** key by itself. The game ends when questions run out or when you press INTERRUPT (**Alt-Pause**); **quiz** reports a score and exits.

Flags

-i file Substitutes the named *file* for the standard index file.

Note: In the following syntax description, brackets are normally used to indicate that an item is optional; a bold-faced bracket or brace, however, should be entered as a literal part of the syntax. A vertical list of items indicates that one and only one must be chosen. The lines in *file* must have the following syntax:

```

line      = category [ :category ] . . .
category  = alternate [ !alternate ] . . .
alternate = [primary]
primary   = character
           [category]
           option
option    = {category}
  
```

In an index file, the first category of each line must specify the name of an information file (the information file contains the names of files with quiz material). The remaining categories specify the order and contents of the data in each line of the information file. The quiz data in an information file follows the same syntax. A \ (backslash) is an escape character which allows you to quote syntactically significant characters or to insert a new-line character (\n) into a line. When either a question or its answer is blank, **quiz** does not ask it. The construct **alab** does not work in an information file. Use **a{b}**.

- t Provides a tutorial. Repeats missed questions and introduce new material gradually.

Examples

1. To start a Latin-to-English quiz:

```
/usr/games/quiz latin english
```

The **quiz** command displays Latin words and waits for you to enter what they mean in English.

2. To start an English-to-Latin quiz:

```
/usr/games/quiz english latin
```

3. To set up a Latin-English quiz, add the following line to the index file:

```
/usr/games/lib/quiz/latin:latin:english
```

This line specifies that the file `/usr/games/lib/quiz/latin` contains information about the categories `latin` and `english`.

You can add new categories to the standard index file, `/usr/games/lib/quiz/index`, or to an index file of your own. If you create your own index file, run the **quiz** command with the `-i file` flag to give it your list of quiz topics.

4. This is a sample information file:

```
cor:heart
sacerdos:priest{ess}
quando:when|since|because
optat:{s}he |it |[desires|wishes]\
|desire|wish
alb[us|alum]:white
```

This information file contains Latin and English words. The `:` (colon) separates each Latin word from its English equivalent. Items enclosed in `{ }` (braces) are optional. A `|` (vertical bar) separates two items when entering either is correct. The `[]` (brackets) group items separated by vertical bars.

The first line accepts only the answer heart in response to the Latin word cor. The second accepts either priest or priestess in response to sacerdos. The third line accepts when, since, or because for quando.

The \ (backslash) at the end of the fourth line indicates that this entry continues on the next line. In other words, the fourth and fifth lines together form one entry. This entry accepts any of the following in response to optat:

she desires	it desires	desire
she wishes	it wishes	wish
he desires	desires	
he wishes	wishes	

If you start a Latin-to-English quiz, then the last line of this sample information file instructs **quiz** to ask you the meaning of albus. If you start an English-to-Latin quiz, then **quiz** displays white and accepts albus, alba, or album for the answer.

If any of the characters {, }, [,], or ! appear in a question item, then **quiz** gives the first alternative of every ! group and displays every optional group. Thus, the English-to-Latin question for the fourth definition in this sample is she desires.

Files

```
/usr/games/lib/quiz/index  
/usr/games/lib/quiz/*
```

Purpose

Performs normal startup initialization.

Syntax

`/etc/rc` —¹

¹ This command is not usually run from the command line.

OL805339

Description

When the **init** process starts up the system in normal operating mode, it runs the command file **/etc/rc** to perform the necessary system initialization, including the enabling of various loggers. If the system is being brought up with no file system checking, **init** passes the argument **m** to **rc**. If **init** determines that the root file system needs consistency checking, it passes the argument **d** to **rc**.

The contents of **/etc/rc** may be installation specific, but there are a few things that it should do:

- Check the default file systems if **rc** is passed the **init -d** flag (Run **fsck**)
- Mount the default file systems (Run **mount**)
- Purge temporary files
- Start SNA and Distributed Services (Run **/etc/rc.sna** and **/etc/rc.ds**)
- Set printer defaults
- Enable default ports (Run **pstart**)
- Determine whether to set up stand-alone or active-service system (Run **chngstate**)
- Start the error daemon and run **/etc/rc.include**.

If all of the necessary operations complete successfully, the file exits with a zero return code that allows **init** to start loggers to complete normal initialization and startup.

Notes:

1. Many system daemons such as **cron** are started by **rc** indirectly when it runs **/etc/rc.include**.
2. The mail facility is started by **rc** indirectly when it runs **/etc/rc.include** and **/etc/rc.tcpip**.
3. The root file system is implicitly mounted.

Files

<code>/etc/rc.ds</code>	Performs functions required to start Distributed Services.
<code>/etc/rc.include</code>	Performs functions required to start most program daemons.
<code>/etc/rc.sna</code>	Performs functions required to start SNA.
<code>/etc/rc.tcpip</code>	Performs functions required to start TCPIP.

Related Information

The following commands: “**chgstate**” on page 164, “**cron**” on page 220, “**fsck**, **dfsc**” on page 445, “**init**” on page 521, “**mount**” on page 669, and “**pstart**, **penable**, **pshare**, **pdelay**” on page 791.

The discussion of starting up the system in *Managing the AIX Operating System*.

rcvdist

rcvdist

Purpose

Sends a copy of incoming messages to additional recipients.

Syntax

`/usr/lib/mh/rcvdist` 

`/usr/lib/mh/rcvdist` 

AJ2FL234

Description

The **rcvdist** command is used to forward copies of incoming messages to other users. **rcvdist** is not designed to be run directly by the user; it is designed to be called by `/usr/lib/mh/slocal`. The **rcvdist** command is part of the MH (Message Handling) package.

The **rcvdist** command sends a copy of the incoming message to the specified users. **rcvdist** uses the format string facility described in **mh-format**. You can run **rcvdist** on all incoming messages by specifying the **rcvdist** command in the **.maildelivery** file.

Flag

-help Displays help information for the command.

Files

<code>\$HOME/.maildelivery</code>	The user's local mail delivery instructions.
<code>\$HOME/.forward</code>	The user's default message filter.

Related Information

Other commands: “**ali**” on page 48, “**rcvpack**” on page 810, “**rcvstore**” on page 812, “**rcvttty**” on page 815, “**sendmail**” on page 897, “**slocal**” on page 954, and “**whom**” on page 1222.

The **mh-alias**, **mh-format**, **mh-mail**, and **mh-profile** files in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

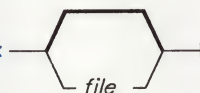
rcvpack

rcvpack

Purpose

Saves incoming messages in a packed file.

Syntax

`rcvpack` — 

`rcvpack` — `-help` —

AJ2FL235

Description

The **rcvpack** command is used to place incoming messages in a packed file. **rcvpack** is not designed to be run directly by the user; it is designed to be called by `/usr/lib/mh/slocal`. The **rcvpack** command is part of the MH (Message Handling) package.

The **rcvpack** command appends a copy of the incoming message to the specified file and runs the **packf** command on the file. You can run **rcvpack** on all incoming messages by specifying the **rcvpack** command in the `.maildelivery` file.

Flag

-help Displays help information for the command.

Files

`$HOME/.maildelivery`
`$HOME/.forward`

The user's local mail delivery instructions.
The user's default message filter.

Related Information

The following commands: **inc** on page 518, **packf** on page 733, **rcvdist** on page 808, **rcvstore** on page 812, **rcvttty** on page 815, **sendmail** on page 897, and **slocal** on page 954.

The **mh-alias**, **mh-format**, **mh-mail**, and **mh-profile** files in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

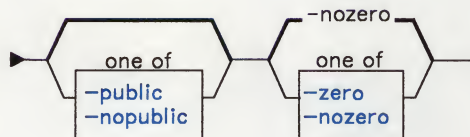
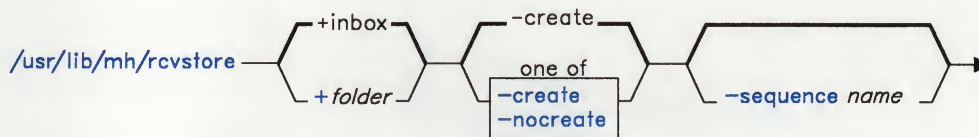
rcvstore

rcvstore

Purpose

Incorporates new mail from standard input into a folder.

Syntax



`/usr/lib/mh/rcvstore` `— -help —`

AJ2FL237

Description

The `rcvstore` command is used to incorporate incoming messages. `rcvstore` is not designed to be run directly by the user; it is designed to be called by `/usr/lib/mh/slocal`. The `rcvstore` command is part of the MH (Message Handling) package.

The `rcvstore` command accepts messages from standard input and places them in a specified folder. You can run `rcvstore` on all incoming messages by specifying the `rcvstore` command in the `.maildelivery` file.

You can specify `rcvstore` flags in `$HOME/.maildelivery` or as with most MH commands, in `$HOME/.mh-profile`.

Flags

-create	Creates the specified folder in your mail directory if the folder does not exist.
+folder	Places the incorporated messages in the specified folder. The default is +inbox .
-help	Displays help information for the command.
-nocreate	Does not create the specified folder if the folder does not exist.
-npublic	Restricts the specified sequence to your usage. -npublic does not restrict the messages in the sequence, only the sequence. This flag is the default if the folder is write-protected from other users.
-nozero	Appends the messages incorporated by rcvstore to the specified sequence (see the -zero flag).
-public	Makes the specified sequence available to other users. -public does not make protected messages available, only the sequence. This flag is the default if the folder is not write-protected from other users.
-sequence name	Adds the incorporated messages to the specified sequence.
-zero	Clears the specified sequence before placing the incorporated messages into the sequence. This flag is the default (see the -nozero flag).

Profile Entries

Folder-Protect:	Sets the protection level for your new folder directories.
Msg-Protect:	Sets the protection level for your new message files.
Path:	Specifies your <i>user-mh-directory</i> .
Unseen-Sequence:	Specifies the sequences used to keep track of your unseen messages.
Rcvstore:	Specifies flags for the rcvstore program.

Files

\$HOME/.mh-profile	The MH user profile.
\$HOME/.maildelivery	The user's local mail delivery instructions.
\$HOME/.forward	The user's default message filter.

Related Information

The following commands: “**inc**” on page 518, “**rcvdist**” on page 808, “**rcvpack**” on page 810, “**rcvttty**” on page 815, “**sendmail**” on page 897, and “**slocal**” on page 954.

The **mh-alias**, **mh-format**, **mh-mail**, and **mh-profile** files in *AIX Operating System Technical Reference*.

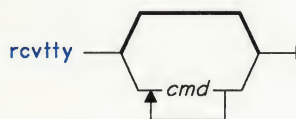
“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

rcvttty

Purpose

Notifies the user of incoming messages.

Syntax



rcvttty — -help —

AJ2FL238

Description

The **rcvttty** command is used to send the user a message when incoming mail has arrived. **rcvttty** is not designed to be run directly by the user; it is designed to be called by **/usr/lib/mh/slocal**. The **rcvttty** command is part of the MH (Message Handling) package.

The **rcvttty** command sends a one-line scan listing to your terminal. If you give **rcvttty** a command as an argument, **rcvttty** executes the command with the incoming message as the command's standard input, and sends the output to the terminal. For **rcvttty** to write output to your terminal, your terminal's write permission must be set to "All".

You can run **rcvttty** on all incoming messages by specifying the **rcvttty** command in the **.maildelivery** file.

Flag

-help Displays help information for the command.

Files

\$HOME/.maildelivery	The user's local mail delivery instructions.
\$HOME/.forward	The user's default message filter.

Related Information

The following commands: “**rcvdist**” on page 808, “**rcvpack**” on page 810, “**rcvstore**” on page 812, “**rcvttty**” on page 815, “**sendmail**” on page 897, and “**slocal**” on page 954.

The **mh-alias**, **mh-format**, **mh-mail**, and **mh-profile** files in *AIX Operating System Technical Reference*.

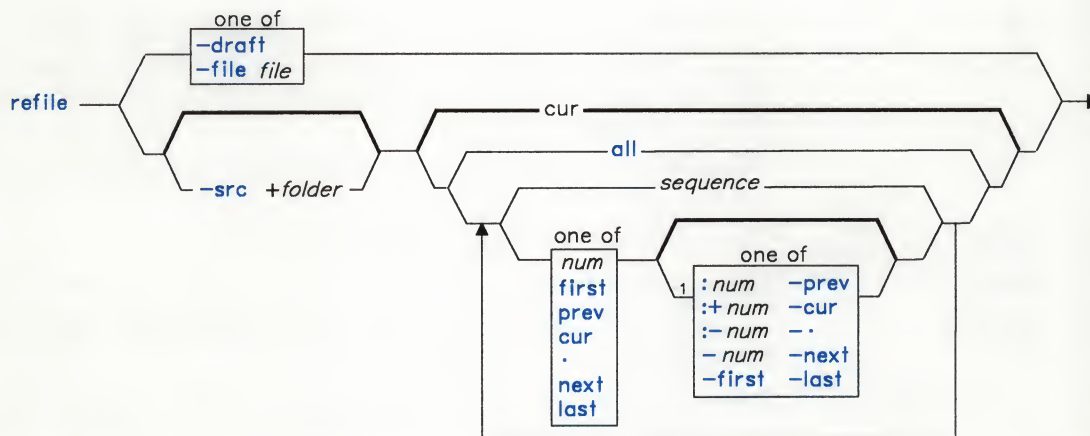
“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

refile

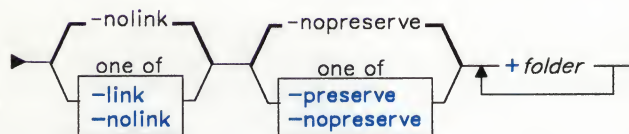
Purpose

Files messages in other folders.

Syntax



AJ2FL223



refile — -help —|

AJ2FL169

¹ Do not put a blank between these items.

OL805308

Description

The **refile** command is used to copy and move messages to other files. The **refile** command is part of the MH (Message Handling) package and can be used with MH and AIX commands.

The **refile** command copies messages or moves messages from one folder to another folder. If a destination folder does not exist, **refile** asks if it should create the folder.

Flags

-draft	Copies the current draft message from your mail directory.
-file file	Copies the specified file. The file must be in valid message format. (Use the inc command to incorporate new messages and to format them correctly.)
+folder	Copies the messages to the specified folder. Any number of folders can be specified.
-help	Displays help information for the command.
-link	Leaves the messages in the source folder or file after they are copied.
-nolink	Removes the messages from the source folder or file after they are copied. This flag is the default.
-nopreserve	Renumbers the messages that are copied. Renumbering begins with the number that is one higher than the last message in the destination folder. This is the default.
-preserve	Preserves the message numbers of the messages that are copied. If messages with those numbers already exist, refile issues an error message and does not alter the contents of the folders.
-src +folder msgs	Specifies the messages to be copied. You can use the following message references when specifying <i>msgs</i> :

<i>num</i>	first	prev
cur	.	next
last	all	<i>sequence</i>

The default message is the current message in the current folder. If a folder is specified, it becomes the current folder. If the **-link** flag and **all** are used, the current message does not change. Otherwise, if a message is specified, that message becomes the current message.

Profile Entries

Current-Folder:	Sets your default current folder.
Folder-Protect:	Sets the protection level for your new folder directories.
Path:	Specifies your <i>user-mh-directory</i> .
rmmproc:	Specifies the program used to remove messages from a folder.

Files

<code>\$HOME/.mh-profile</code>	The MH user profile.
---------------------------------	----------------------

Related Information

Other MH commands: “**folder**” on page 429, “**folders**” on page 433.

The **mh-profile** file in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

regcmp

regcmp

Purpose

Compiles patterns.

Syntax



OL805211

Description

The **regcmp** command compiles the pattern in *file*, placing its output in *file.i*.

In most cases, **regcmp** makes unnecessary the use of the **regcmp** system call in your C programs, saving execution time and program size. The output of **regcmp** is C source code. Make each file entry a C variable name, followed by one or more blanks, followed by a pattern enclosed in double quotation marks (" "). Compiled patterns are initialized **char** declarations. Thus, *file.i* can be included in C programs, and *file.c* can be a file parameter to the **cc** command. The C program that uses **regcmp** output should use the **regex** subroutine to apply it to a string. (See **regcmp** and **regex** in *AIX Operating System Technical Reference*.)

Flag

- Places the output in *file.c*

Related Information

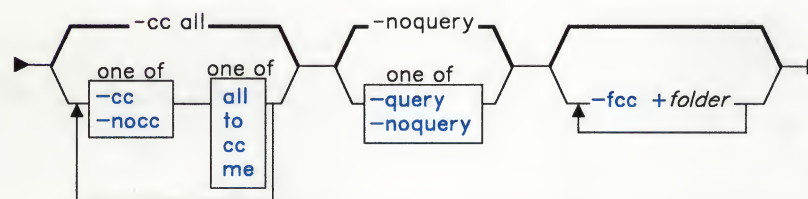
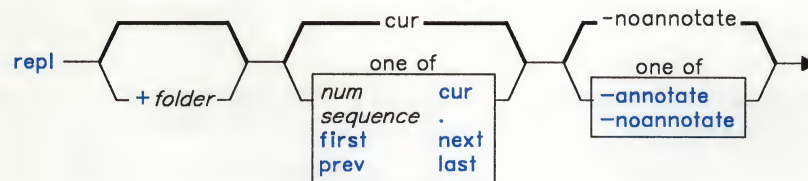
The **regcmp** subroutine in *AIX Operating System Technical Reference*.

repl

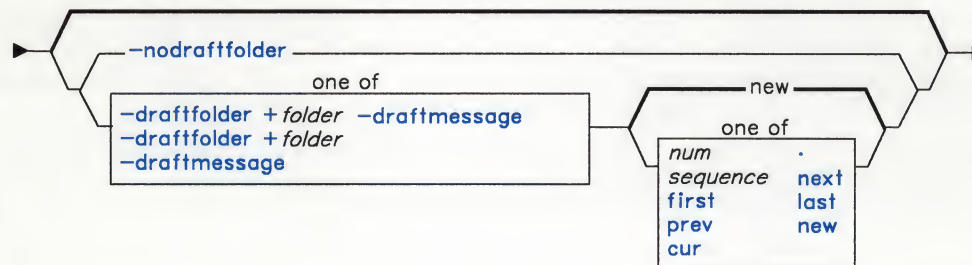
Purpose

Replies to a message.

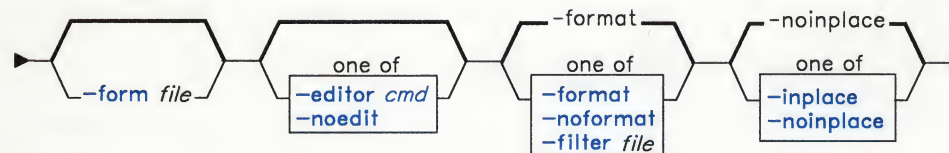
Syntax



AJ2FL240

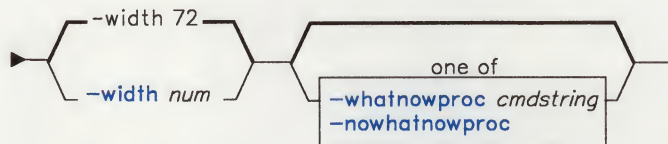


AJ2FL157



AJ2FL241

repl



repl — -help —

AJ2FL226

Description

The **repl** command is used to compose a reply to a message. **repl** is part of the MH (Message Handling) package and can be used with other MH and AIX commands.

By default, **repl** copies a message form to a new draft message and invokes an editor. You can then fill in the message header fields **To:** and **Subject:**, fill in or delete the other header fields (such as **cc:** and **Bcc:**), and add the body of the message. When you exit the editor, the **repl** command invokes the MH command **whatnow**. You can press **Enter** to see a list of the available **whatnow** subcommands. These subcommands enable you to continue editing the reply, list the reply, direct the disposition of the reply, or end the processing of the **repl** command. “**whatnow**” on page 1215 describes the subcommands.

You can specify the message that you want to reply to by using the **+folder msg** flag. If you do not specify a message, **repl** replies to the current message.

You can specify the format of the reply by using the **-form** flag. If you do not specify this flag, **repl** uses your default message format located in the file *user_mh_directory/replcomps*. If this file does not exist, **repl** uses the system default message format located in */usr/lib/mh/replcomps*.

Note: The line of dashes or a blank line must be left between the header and the body of the message for the message to be identified when it is sent.

Flags

-annotate

Annotates the message being replied to with the lines:

Replied: *date*
Replied: *addrs*

The annotation appears in the original draft message so that you can maintain a complete list of activities associated with the original message.

- If you do not actually send the reply using the immediate **repl** command, the **-annotate** flag may fail to provide annotation. The **-inplace** flag forces annotation to be done in place.
- cc names** Specifies the users who will be listed in the **cc:** field of the reply. You can specify the following for *names*: **all**, **to**, **cc**, and **me**. The default is **-cc all**.
- draftfolder +folder** Places the draft message in the specified folder. If you do not specify this flag, **repl** selects a default draft folder according to the information supplied in the MH profiles. You can define a default draft folder in **\$HOME/.mh-profile**. If **-draftfolder +folder** is followed by *msg*, *msg* represents the **-draftmessage** attribute.
- draftmessage msg** Specifies the draft message. You can use one of the following message references as *msg*:
- | | | |
|-------------|-----------------|--------------|
| <i>num</i> | <i>sequence</i> | first |
| prev | cur | . |
| next | last | new |
- The default draft message is **new**.
- editor cmd** Specifies that *cmd* is the initial editor for composing the reply. If you do not specify this flag, **repl** selects a default editor or suppresses the initial edit, according to the information supplied in the MH profiles. You can define a default initial editor in **\$HOME/.mh-profile**.
- fcc +folder** Places a file copy of the reply in *folder*. If you do not specify this flag, **repl** will not produce a file copy.
- filter file** Reformats the message being replied to and places the reformatted message in the body of the reply. **-filter** uses the **mhl** command and the specified format file. If you do not specify this flag, **repl** will omit the original message from the reply. The **repl** command does not have a default filter file. Thus, if you specify **-filter**, you must also specify *file*.
- +folder msg** Replies to the specified message in the specified folder. You can use one of the following message references as *msg*:
- | | | |
|-------------|-----------------|--------------|
| <i>num</i> | <i>sequence</i> | first |
| prev | cur | . |
| next | last | |
- The default message is the current message in the current folder. If you specify a folder, that folder becomes the current folder. The message being replied to becomes the current message.

repl

-form <i>file</i>	Uses the form contained in the specified file for the form of the reply. repl treats each line in <i>file</i> as a format string.
-format	Removes duplicate addresses from the fields To: , cc: , and Bcc: and standardizes these fields. repl also uses the columns specified by the -width flag to determine the format of these fields. This flag is the default.
-help	Displays help information for the command.
-inplace	Forces annotation to be done in place in order to preserve links to the annotated message.
-noannotate	Does not annotate the message. This flag is the default.
-nocc <i>names</i>	Specifies the users who will not be listed in the cc: field of the reply. You can specify the following for <i>names</i> : all , to , cc , and me .
-nodraftfolder	Places the draft in the file <i>user_mh_directory/draft</i> .
-noedit	Suppresses the initial edit.
-noformat	Does not remove duplicate addresses from the fields To: , cc: , and Bcc: or standardize these fields. repl also does not use the columns specified by the -width flag to determine the format of these fields.
-noinplace	Does not perform annotation in place. This flag is the default.
-noquery	Automatically builds the To: and cc: fields. This flag is the default.
-nowhatnowproc	Does not invoke a program that guides you through the reply tasks. The -nowhatnowproc flag also prevents any edit from occurring.
-query	Builds the To and cc: fields by interactively asking you if you want each address that would normally be placed in these fields actually placed in these fields.
-whatnowproc <i>cmdstring</i>	Invokes <i>cmdstring</i> as the program to guide you through the reply tasks. See “ whatnow ” on page 1215 for information about the default whatnow program and its subcommands. Note: If you specify <i>whatnow</i> for <i>cmdstring</i> , repl invokes an internal whatnow procedure rather than a program with the file name whatnow .
-width <i>num</i>	Sets the width of the address fields. The default is 72 columns.

Profile Entries

Alternate-Mailboxes:	Specifies your mailboxes.
Current-Folder:	Sets your default current folder.
Draft-Folder:	Sets your default folder for drafts.
Editor:	Sets your default initial editor.
fileproc:	Specifies the program used to refile messages.
mhlproc:	Specifies the program used to filter the message for which you are creating a reply.
Msg-Protect:	Sets the protection level for your new message files.
Path:	Specifies your <i>user-mh-directory</i> .
whatnowproc:	Specifies the program used to prompt What now? questions.

Files

<i>/usr/lib/mh/replcomps</i>	The MH default reply template.
<i>user-mh-directory/replcomps</i>	The user's default reply template. (If it exists, it overrides the MH default reply template.)
<i>\$HOME/.mh-profile</i>	The MH user profile.
<i>user-mh-directory/draft</i>	The draft file.

Related Information

Other MH commands: “**ali**” on page 48, “**anno**” on page 50, “**comp**” on page 185, “**dist**” on page 336, “**forw**” on page 438, “**mhl**” on page 643, “**prompter**” on page 778, “**send**” on page 893, “**whatnow**” on page 1215, “**whom**” on page 1222.

The **mh-alias**, **mh-format**, **mh-mail**, and **mh-profile** files in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

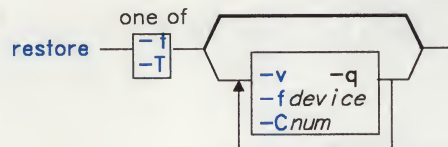
restore

restore

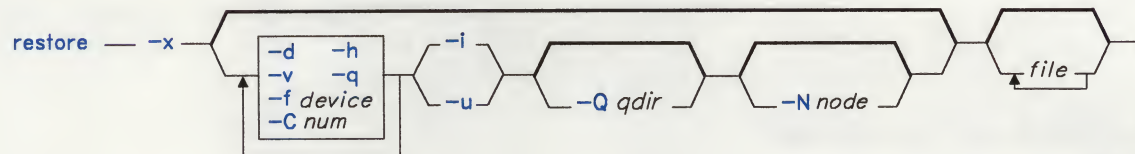
Purpose

Copies back files created by the **backup** command.

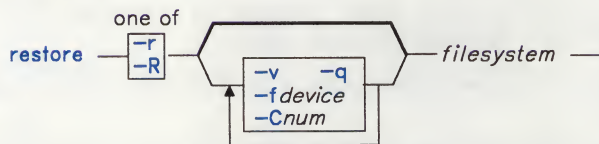
Syntax



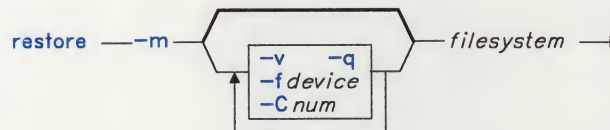
OL805251



OL805352



OL805353



OL805441

Description

The **restore** command reads files written by the **backup** command to a backup medium and restores them to a file system. You can restore files that are backed up on a local system or on a remote system.

There are four ways to use the **restore** command:

- To display a table of contents for the backup (**-T**) or to display label information (**-t**)
- To restore specified files (**-x**)
- To restore an entire file system (**-r**) or begin at an arbitrary volume number (**-R**).
- To restore an entire minidisk (**-m**).

When you do not specify a restore device, the **restore** command reads files from a default backup device. For restore by name, **restore -x**, the system reads from **/dev/rfd0** unless you specify a device with the **-f** flag. For restore by file system, **restore -i**, or restore by minidisk, **restore -m**, if **/etc/filesystems** contains a stanza that matches the name you specified and a stanza with a **backupdev** entry, then the system reads from the device specified by **backupdev**. Otherwise, the system reads from **/dev/rmt0** or the *device* specified with the **-f** flag.

If neither **-i** nor **-u** is specified, files are restored to the local node in the current directory. If either **-i** or **-u** is specified, the system needs to know where to restore the files. Either a target node or a qualifying directory or both can be used to tell the system where to restore the files. If they are not specified, **restore** looks for the information in the header file of the backup. If you are restoring files backed up with an old version of the **backup** that does not contain a header and you do not specify a target node and a qualifying directory on the command line, the **restore** command ends in an error.

Notes:

1. If you restore by file system or by minidisk, the source and target must be on the local system. To restore from a remote system, restore by name with the **-i** or **-u** flags. These flags allow users in a distributed services environment to restore files backed up on a remote tape drive.
2. If the file system you are restoring is mounted and is not the root file system, **restore** unmounts the file system before it performs an i-node or a minidisk restore and then remounts the file system before quitting. If the file systems you are restoring include the root file system, **restore** ensures that the other file systems are not in use. If one is, it warns you of this and quits.
3. Files must be restored using the same method by which they were backed up. For example, if a file system was backed up by minidisk, it must be restored by minidisk.
4. When more than one diskette is required, **restore** reads the one mounted, prompts the user for a new one, and waits for the user's response (unless you are in unattended mode). After inserting the new diskette, press **Enter** to continue restoring files.

Flags

- Cnum** Specifies the number of blocks to read in a single input operation. If you do not specify this flag, **restore** selects a default value appropriate for the physical device you have selected. Larger values of *num* result in longer physical transfers from tape devices. **restore** always ignores the value of the **-C** flag

when it reads a diskette; the input is always read in clusters that occupy a complete track.

- d Indicates that if *file* is a directory, all files in that directory should be restored. In this case, the name of each restored file is always its name as shown by **restore -T**, whether the backup was by name or by i-node. The *file* names supplied need not be directories. Thus, for i-node backups:

```
restore -x a/b/file.c
```

creates a file whose name is its i-node number, while:

```
restore -xd a/b/file.c
```

creates a file named `a/b/file.c`. With this flag, *file* names can include pattern-matching characters, although you must quote these characters to prevent their expansion by the shell.

Use this flag only when you are restoring by individual file name (-x).

- fdevice Specifies the input device. Specify *device* as a file name (such as `/dev/rmt0`) to get input from the named device or specify - (minus) to get input from the standard output device. The - feature enables you to improve performance when restoring from streaming tape by piping the output of a **dd** command to the **restore** command (see example). The **restore** command recognizes a special syntax for the names of input files. If the device parameter is a range of names, for example `/dev/rfd0-3`, **restore** automatically goes from one drive in the range to the next. After using all of the specified drives, it stops and requests that another diskette be inserted.

- h Specifies that the access and modification times of restored files are to be set to the time of restoration. (The default action is to set the access and modification times to the file times on the backup medium.) If a restored file is an archive, the modification times in all the member headers are also set to the time of restoration. You can specify this flag only when you are restoring individually named files.

- i Enables users in a distributed services environment to restore from a backup medium on a remote system in interactive mode (user input is permitted).

- m Restores an entire minidisk as an exact image.

Note: You can use this flag only with minidisks that are at least as large as the original minidisk that was backed up. If the minidisk is larger than the original, the leftover space becomes unusable after restoring the minidisk. You can use **restore -t** to see how large a minidisk you need.

- N node Specifies the node on which to restore files. The *node* can be a node nickname or a node id. The **restore** command uses this node instead of the node in the backup header.

- q Specifies that the removable medium is ready to use. In this case, **restore** proceeds without prompting you to prepare the removable medium.
- Q *qdir* Specifies the qualifying directory in which to restore files. The *qdir* can be a relative or absolute directory. The **restore** command uses this qualifying directory instead of the directory in the backup header. Current directory relative names extracted from the backup medium are placed in this directory.
- r Restores an entire file system. Use this flag with i-node backups only (see “**backup**” on page 88). *filesystem* can be a device name (block or character special file) or a directory name that **restore** looks up in */etc/filesystems*.

If you are restoring a *full* (level 0) **backup**, run the **mkfs** command to create an empty file system before doing the restore. If you are restoring an *incremental backup* at, for example, level 2, run **mkfs**, then restore the appropriate level 0 backup, then the level 1 backup, and finally the level 2 backup.

Warning: If you do not follow this procedure carefully, you can ruin an entire file system. As an added safety precaution, run **fsck** after you restore each backup level.
- R Restarts an aborted **restore** at a specified point. **restore** prompts you for the starting volume number. This flag is invalid in combination with the **-m** flag.
- T Displays the backup file header and the names of the backed up files. If the backup was made by name (**backup -i**), the names displayed are the ones you provided to **backup**. If the backup was made by i-node, **restore** displays the i-number of each file along with the file name. The names are relative to the root directory of the file system backed up. The only exception is the root directory itself, whose name is given as a slash (/).
- t Displays only the backup file header.
- u Enables users to restore files in unattended mode (user input is not permitted) from a backup medium on a remote system. If any user input (such as *Please mount volume 1 on /dev/rfd0*) is required, the command ends in an error. This enables users to set up a shell file that restores files at night or at other times when a user is unavailable.
- v Reports the progress of the restoration as it proceeds.
- x Restores individually named files. The names must be in the same form as the names shown by **restore -T**. With a name backup, **restore** gives the restored file whatever name was supplied when the file was backed up. If the original name was specified relative to the current directory, **restore** creates a file relative to the current directory. **restore** automatically creates any needed directories. With an i-node backup, the name of the restored file is the same as its i-number. This flag is invalid with the **-m** flag and the **-r** flag.

Examples

1. To list the names of files previously backed up:

```
restore -T
```

Information is read from the default backup device **/dev/rfd0**. If individual files were backed up, then only the file names are displayed. If an entire file system was backed up, the i-number is also shown.

2. To display technical information about a backup:

```
restore -t
```

This command displays information including when the backup was made, which file system was saved, and whether it is a backup by name, a backup by minidisk, or a backup by file system or i-node.

3. To restore files to the main file system:

```
restore -x -v
```

The **-x** extracts all the files from the backup medium and restores them to their proper places in the file system. The **-v** displays a progress report as each file is restored. If a file system backup is being restored, then the files are named with their i-numbers. Otherwise, just the names are displayed.

4. To copy selected files:

```
restore -xv /u/jim/manual/chap1
```

This command extracts the file **/u/jim/manual/chap1** from the backup medium and restores it. To work properly, **/u/jim/manual/chap1** must be a name that can be displayed by **restore -T**.

5. To copy all the files in a directory:

```
restore -xdv manual
```

This command restores the directory **manual** and the files in it. If it does not exist, a directory named **manual** is created in the current directory to hold the files being restored.

6. To restore an entire file system backup:

```
mkfs /dev/hd1  
restore -rv /dev/hd1
```

This command restores an entire file system backup onto **/dev/hd1**. It destroys and replaces any file system that was previously stored on **/dev/hd1**. If the backup was made using incremental file system backups, restore the backups in increasing backup-level order (0, 1, 2 . . .).

7. To restore a minidisk:

```
restore -m /dev/hd1
```

This restores the exact image of minidisk /dev/hd1. You can also identify the minidisk by its stanza name in the `/etc/filesystems` file.

8. To restore files in interactive mode from the remote default device specifying a new target node and qualifying directory:

```
restore -xi -N nick -Q /u/nick
```

This command extracts the files from the default remote backup device and restores them to the node `nick`. Any unqualified names from the media are extracted relative to the directory `/u/nick`. The contents of the backup header (if a header exists) is ignored.

9. To restore files in unattended mode specifying a target node id:

```
restore -xuN 10813661
```

Files from the default backup device are restored at the node whose node ID is 10813661. Since a qualifying directory is not specified and the backup contains a backup header file, the **restore** command extracts the files to the qualifying directory specified in the header.

10. To improve performance on streaming tape, pipe the **dd** command to the **restore** command:

```
dd of=/dev/rmt0 bs=30b | restore -xf-
```

The **dd** command copies the files from an output file which is a streaming tape device (`of=/dev/rmt0`) and specifies a file size of 30 blocks (`bs=30b`). The output is piped to **restore**. The **restore** command gets the input from the standard input device (`f-`) and restores up by name (`x`).

Files

<code>/etc/filesystems</code>	Descriptions of mountable file systems; consulted for default parameters.
<code>/dev/rfd0</code>	Default restore device.

Related Information

The following command: “**backup**” on page 88.

The **file systems** and **backup files** in *AIX Operating System Technical Reference*.

“Backing up and Restoring Files” in *Using the AIX Operating System*.

rexd

rexd

Purpose

Handles remote execution requests.

Syntax

`/usr/etc/rpc.rexd` —|

OL805507

Description

When a remote execution request is made, the **inetd** daemon starts **rex**d if the appropriate entry is in the **/etc/inetd.conf** file.

Noninteractive programs use standard file descriptors connected directly to Simulates phototypesetter output for a Tektronix 4014 work station connections. Interactive programs use pseudo-terminals, similar to the login sessions provided by **rlogin**. Diagnostic messages are normally displayed on the console and returned to the requestor.

The **rex**d daemon can use NFS to mount file systems specified in the remote execution request.

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

Files

<code>/dev/ptsn</code>	Pseudo-terminals used for interactive mode.
<code>/etc/passwd</code>	List of authorized users.
<code>/etc/inetd.config</code>	TCP/IP configuration file.

Related Information

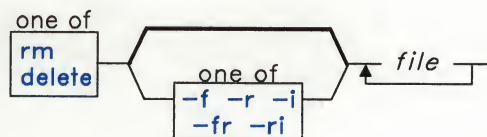
The following command: “**on**” on page 726.

rm

Purpose

Removes files or directories.

Syntax



OL805212

Description

The **rm** (**delete**) command removes the entries for *files* from a directory. If an entry is the last link to a file, it is destroyed. To remove a file, you must have write permission in its directory, but neither read nor write permission for the file itself if you own it or are acting with superuser authority.

If a file has no write permission and standard input is a work station, **rm** displays the file permission code and reads a line from standard input. If that line begins with **y**, **rm** deletes the file. If the response is anything other than **y**, **rm** does nothing.

Japanese Language Support Information

An affirmative response in Japanese Language Support matches one of the elements in the environment variable **YESSTR**.

Flags

- f** Does not prompt before removing a write-protected file.
- i** Prompts you before deleting each file. When you use both **-i** and **-r** together, **rm** also asks if you want to examine directories.
- r** Permits recursive removal of directories and their contents (for cases where *file* is a directory).

Examples

1. To delete a file:

```
rm myfile
```

If there is another link to this file, then the file remains under that name, but the name `myfile` is removed. If `myfile` is the only link, the file itself is deleted.

2. To delete a file silently:

```
rm -f core
```

This removes `core` without asking any questions or displaying any error messages. This is normally used in shell procedures. It prevents confusing messages from being displayed when deleting files that may or may not exist.

3. To delete files one by one:

```
rm -i mydir/*
```

This interactively asks you if you want to remove each file. After each file name is displayed, enter `y` to delete the file, or press **Enter** to keep it.

Japanese Language Support Information

Enter one of the allowed affirmative responses at the prompts. The allowed affirmative responses are defined in the environment variable **YESSTR**.

4. To delete a directory tree:

```
rm -ir manual
```

This recursively removes the contents of all subdirectories of `manual`, then removes `manual` itself, asking if you want to remove each file. For example:

```
You: rm -ir manual
System: directory manual:
You: y
System: directory manual/draft1:
You: y
System: manual/draft1/chapter1:
You: y
System: manual/draft1/chapter2:
You: y
System: manual/draft1:
You: y
System: directory manual/draft2
```



```
You: y
System: manual/draft2:
You: n
System: manual:
You: y
```

Here, **rm** first asks if you want it to search the directory `manual`. Because `manual` contains directories, **rm** next asks for permission to search `manual/draft1` for files to delete, and then asks if you want it to delete the files `manual/draft1/chapter1` and `manual/draft1/chapter2`. **rm** next asks for permission to search the directory `manual/draft2`, and then asks for permission to delete the directories `manual/draft1`, `manual/draft2`, and `manual`. Because you denied permission to remove `manual/draft2`, **rm** will not remove `manual`. Instead, you will see the message `rmdir: manual not empty`.

Related Information

The following commands: “**del**” on page 308 and “**ln**” on page 581.

The **unlink** system call in *AIX Operating System Technical Reference*.

The discussion of Japanese Language Support in *Japanese Language Support User's Guide*.

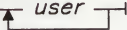
rmail

rmail

Purpose

Handles remote mail received via uucp.

Syntax

`rmail` 

AJ2FL255

Description

The **rmail** command interprets incoming mail received via **uucp**. It collapses From lines in the form generated by **bellmail** into a single line of the form:

return-path!sender

It passes the processed mail on to **sendmail**.

The **rmail** command works with **uucp** and **sendmail**. This is a new command with Version 2.2 of AIX Operating System. It is not the same as the **rmail** command found in earlier versions of AIX Operating System.

Related Information

The following commands: “**uucp**” on page 1144 and “**sendmail**” on page 897.

rm~~del~~

Purpose

Removes a delta from a Source Code Control System (SCCS) file.

Syntax

```
rmdel — -rSID file
```

OL805213

Description

The **rm~~del~~** command removes the delta specified by *SID* from each named **Source Code Control System** (SCCS) *file*. You can remove only the most recently created delta in a branch, or the latest trunk delta if it has no branches. In addition, the *SID* you specify must not be a version currently being edited for the purpose of making a delta. To remove a delta, you must either own the SCCS file and the directory, or you must be the user who created the delta you want to remove.

If you specify a directory in place of *file*, **rm~~del~~** performs the requested actions on all SCCS files (those with file names that have the **s.prefix**). If you specify a - (minus) in place of *file*, **rm~~del~~** reads standard input, and interprets each line as the name of an SCCS file. **rm~~del~~** continues to take input until it reads an end-of-file character.

Flag

-rSID Removes the delta *SID* from the SCCS file. This flag is required.

Related Information

The following commands: “**delta**” on page 310, “**get**” on page 477, “**help**” on page 513, and “**prs**” on page 781.

The **sccsfile** file in *AIX Operating System Technical Reference*.

The discussion of SCCS in *AIX Operating System Programming Tools and Interfaces*.


rmkdir

rmkdir

Purpose

Removes a directory.

Syntax

`rmkdir`  *directory*

OL805252

Description

The **rmkdir** command removes a *directory* from the system. The *directory* must be empty before you can remove it, and you must have write permission in its parent directory. Use the **ls -l** command to see if the *directory* is empty.

Example

To empty and remove a directory:

```
rm mydir/* mydir/*
rmkdir mydir
```

This removes the contents of *mydir*, then removes the empty directory. The **rm** command displays an error message about trying to remove the directories **.** (dot) and **..** (dot dot), and then **rmkdir** removes them.

Note that **rm mydir/* mydir/*** first removes files with names that do not begin with a dot, then those with names that do begin with a dot. You may not realize that the directory contains file names that begin with a dot because the **ls** command does not normally list them.

Related Information

The following command: “**rm**” on page 833.

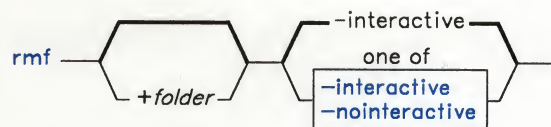
The **unlink** and **rmkdir** system calls in *AIX Operating System Technical Reference*.

rmf

Purpose

Removes a folder.

Syntax



rmf — -help —

AJ2FL165

Description

The **rmf** command is used to remove folders and the messages that they contain. **rmf** is part of the MH (Message Handling) package and can be used with other MH and AIX commands.

Warning: The **rmf** command irreversibly deletes messages that do not have other links.

The **rmf** command deletes all of the messages within the specified folder and then deletes the folder. If the folder contains files that are not messages, **rmf** does not delete those files and reports an error.

If you have read-only access to the specified folder, **rmf** does not delete the folder or any of its messages. **rmf** deletes only your private sequences and your current message information from the profile.

The **rmf** command does not delete folders recursively. Thus, you cannot remove subfolders by requesting the removal a parent folder.

rmf

Flags

- | | |
|-----------------------|---|
| +folder | Specifies the folder to be removed. If you remove a subfolder, the parent of that folder becomes the current folder. If you remove the current folder, +inbox becomes current. The default folder is the current folder. If +folder is not specified and rmf cannot find the current folder, rmf requests confirmation for removing +inbox . |
| -help | Displays help information for the command. |
| -interactive | Requests confirmation before removing the folder. If +folder is not specified, this is the default. |
| -nointeractive | Removes the folder and its messages without requesting confirmation. If +folder is specified, this is the default. |

Profile Entries

- | | |
|------------------------|---|
| Current-Folder: | Sets your default current folder. |
| Path: | Specifies your <i>user_mh_directory</i> . |

Files

- | | |
|---------------------------------|----------------------|
| <code>\$HOME/.mh-profile</code> | The MH user profile. |
|---------------------------------|----------------------|

Related Information

The MH command “**mmm**” on page 841.

The **mh-profile** file in *AIX Operating System Technical Reference*.

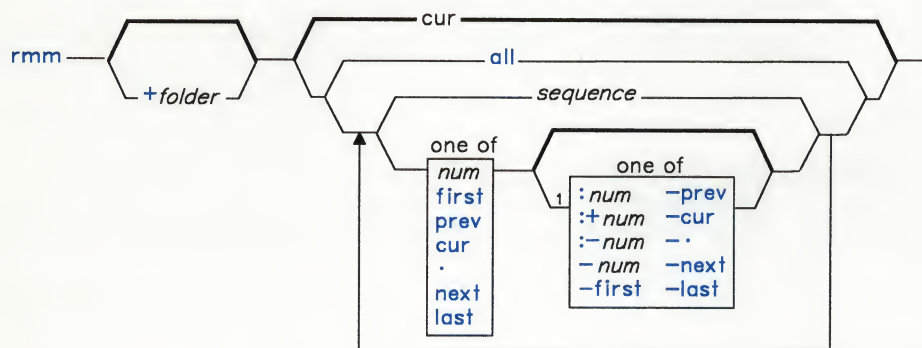
“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

rmm

Purpose

Removes messages.

Syntax



`rmm -help`

AJ2FL202

AJ2FL203

¹ Do not put a blank between these items.

OL805308

Description

The **rmm** command is used to remove messages from active status. **rmm** is part of the MH (Message Handling) package and can be used with other MH and AIX commands.

The **rmm** command renames the specified message files so that their file names have preceding commas. You can use these files as temporary backups and arrange for the **cron** command to delete your backups periodically.

rmm

Flags

+folder msgs Specifies the messages that you want to remove. *msgs* can be several messages, a range of messages, or a single message. You can use the following message references when specifying *msgs*:

<i>num</i>	first	prev
cur	.	next
last	all	<i>sequence</i>

The default message is the current message in the current folder. **rmm** does not change the current message.

-help Displays help information for the command.

Profile Entries

Current-Folder:	Sets your default current folder.
Path:	Specifies your <i>user_mh_directory</i> .
rmmproc:	Specifies the program used to remove messages from a folder.

Files

\$HOME/.mh-profile The MH user profile.

Related Information

The MH command “**rmf**” on page 839.

The **mh-profile** file in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

rpcgen

Purpose

Compiles a Remote Procedure Call program.

Syntax

`rpcgen infile`

`rpcgen -c infile -o outfile`

`rpcgen -h infile -o outfile`

`rpcgen -s transport infile -o outfile`

OL805497

Description

The **rpcgen** command generates C Language code for implementing an RPC protocol. Input to **rpcgen** is in Remote Procedure Call Language (RPCL). RPCL is similar to the C Programming Language.

The **rpcgen** command operates in the following modes:

- Converts RPCL definitions to C Language definitions and puts them in a header file.
- Compiles the XDR routines that serialize or convert the data between the machine issuing the Remote Procedure Call and the machine carrying it out.
- Compiles converted RPCL definitions and puts them in a header file named *infile.h*. Compiles the XDR routines and puts them in *infile.c*.

rpcgen

- Compiles an RPC server skeleton. Using the skeleton, you can write local procedures that implement RPC servers without invoking RPC protocols.

In each mode, the input can contain comments (with the same format as C Language comments) and preprocessor directives. The comments are ignored and the directives are copied into the output header file. You can customize XDR routines by leaving some data types undefined. For every undefined data type, **rpcgen** will assume that a routine with an **xdr-** prefix exists.

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

Flags

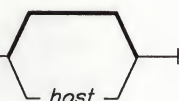
- | | |
|----------------------------|---|
| -c | Compiles XDR routines. |
| -h | Compiles C data definitions in a header file. |
| -o <i>outfile</i> | Specifies the name of the output file. If you do not specify an output file, rpcgen uses the standard output as the default. |
| -s <i>transport</i> | Compiles a server using a specified data transport. This flag can be invoked more than once in order to compile a server that serves multiple transports. |


rpcinfo


Purpose

Reports Remote Procedure Call status information

Syntax

`/etc/rpcinfo -p` 

`/etc/rpcinfo -u` 

`/etc/rpcinfo -t` 

OL805495

Description

The **rpcinfo** command reports the status of Remote Procedure Call services on a specified host. The *host* parameter specifies the Remote Procedure Call server. The *program* parameter specifies the program used by the remote procedure. It can be a name or a number. The version number of the program used by the remote procedure is specified by *version*. If you do not specify a version number, **rpcinfo** searches for all registered version numbers and calls each one.

Note: The **rpcinfo** command uses *version 0* to search for all registered versions, since 0 is not usually assigned as a program's version number. If *version 0* is assigned to a program's version number, **rpcinfo** uses a high level number in its place.

rpcinfo

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

Flags

- p** Queries the *host* portmap service and displays a list of registered RPC programs. If *host* is not specified, the value returned by **hostname** is the default value.
- t** Uses the TCP/IP data transport to make a Remote Procedure Call to **procedure** 0 of the *program* on the specified *host* and report if a response was received.
- u** Use the UDP/IP data transport to make a Remote Procedure Call to **procedure** 0 of the *program* on the specified *host* and report if a response was received.

Files

/etc/rpc RPC program names.
/etc/inetd.conf TCP/IP configuration file.

rstatd

Purpose

Returns NFS performance statistics from the kernel

Syntax

`/usr/etc/rpc.rstatd`—l

OL805494

Description

The **rstatd** daemon returns performance statistics from the kernel. The **inetd** daemon invokes **rstatd**.

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

File

`/etc/inetd.config` TCP/IP configuration file.

Related Information

The following command: “**rup**” on page 854.

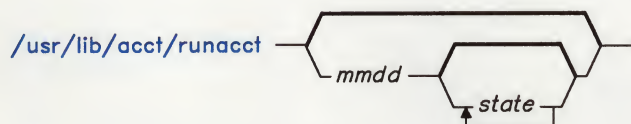
runacct

runacct

Purpose

Runs daily accounting.

Syntax



OL805253

Description

The **runacct** command is the main daily accounting shell procedure. Normally initiated by **cron**, **runacct** processes connect, fee, disk, queueing system and process accounting data files. It also prepares summary files for the **prdaily** procedure or for billing purposes.

The **runacct** command protects active accounting files and summary files in the event of run-time errors. It records its progress by writing descriptive messages into the file **/usr/adm/acct/nite/active**. When **runacct** encounters an error, it writes a diagnostic message to **/dev/console**, sends **mail** to users **root** and **adm**, and exits.

The **runacct** procedure also creates two temporary files, **lock** and **lock1** in the directory **/usr/adm/acct/nite**, which it uses to prevent two simultaneous calls to **runacct**. It uses the file **lastdate** (in the same directory), to prevent more than one invocation per day.

The **runacct** command breaks its processing into separate, restartable **states**. As it completes each state, it writes the name of the next state in **/usr/adm/acct/nite/statefile**. **runacct** processes the various states in the following order:

State	Actions
SETUP	Moves the active accounting files to working files and restarts the active files.
WTMPFIX	Verifies the integrity of the wtmp file, correcting date changes if necessary.
CONNECT1	Calls acctcon1 to produce connect session records.

CONNECT2	Converts connect session records into total accounting records (tacct.h format).
PROCESS	Converts process accounting records into total accounting records (tacct.h format).
MERGE	Merges the connect and process total accounting records.
FEES	Converts the output of chargefee into total accounting records (tacct.h format) and merges them with the connect and process total accounting records.
DISK	Merges disk accounting records with connect, process, and fee total accounting records.
QUEUEACCT	Sorts the queue (printer) accounting records, converts them into total accounting records (tacct.h format), and merges them with other total accounting records.
MERGETACCT	Merges the daily total accounting records in daytacct with the summary total accounting records in /usr/adm/acct/sum/tacct .
CMS	Produces command summaries in the file /usr/adm/acct/sum/cms .
USEREXIT	If the shell file /usr/adm/siteacct exists, calls it at this point to perform site-dependant processing.
CLEANUP	Deletes temporary files and exit.

To restart **runacct** after a failure, first check the **/usr/adm/acct/nite/active** file for diagnostic messages, then fix any damaged data files such as **pacct** or **wtmp**. Remove the **lock** files and **lastdate** file (all in the **/usr/adm/acct/nite** directory), before restarting **runacct**. You must specify the **mmdd** parameter if you are restarting **runacct**. It specifies the month and day for which **runacct** is to rerun the accounting. **runacct** determines the entry point for processing by reading **statefile**. To override this default action, specify the desired **state** on the **runacct** command line. For a more detailed discussion of restarting **runacct**, see *Managing the AIX Operating System*.

It is not usually a good idea to restart **runacct** in the **SETUP state**. Instead, perform the setup actions manually and restart accounting with the **WTMPFIX** state, as follows:

```
runacct mmdd WTMPFIX
```

If **runacct** fails in the **PROCESS** state, remove the last **ptacct** file, because it will be incomplete.

runacct

Japanese Language Support Information

This command has not been modified to support Japanese characters.

Examples

1. To start **runacct**:

```
nohup /usr/lib/acct/runacct 2> /usr/adm/acct/nite/accterr &
```

This starts **runacct** in the background (&), ignoring all INTERRUPT and QUIT signals (**nohup**). All standard error output is written to the file `/usr/adm/acct/nite/accterr`.

2. To restart **runacct**:

```
nohup /usr/lib/acct/runacct 0601 2>> /usr/adm/acct/nite/accterr &
```

This restarts **runacct** for the day of June 1 (0601). **runacct** reads the file `/usr/adm/acct/nite/statefile` to find out the state to begin with. Standard error output is added to the end of the file `/usr/adm/acct/nite/accterr`.

3. To restart **runacct** in a specific state, in this case the **MERGE** state:

```
nohup /usr/lib/acct/runacct 0601 MERGE 2>> /usr/adm/acct/nite/accterr &
```

Files

<code>/usr/adm/wtmp</code>	Login/logoff history file.
<code>/usr/adm/pacct*</code>	Process accounting file.
<code>/usr/adm/acct/nite/daytacct</code>	Disk usage accounting file.
<code>/usr/adm/qacct</code>	Active queue accounting file.
<code>/usr/adm/fee</code>	Record of fees charge to users.
<code>/usr/adm/acct/sum/*</code>	Command and total accounting summary files.
<code>/usr/adm/acct/nite/ptacct*.mdd</code>	Concatenated version of pacct files.
<code>/usr/adm/acct/nite/active</code>	runacct message file.
<code>/usr/adm/acct/nite/lock*</code>	Prevent simultaneous invocation of runacct .
<code>/usr/adm/acct/nite/lastdate</code>	Contains last date runacct was run.
<code>/usr/adm/acct/nite/statefile</code>	Contains current state to process.

Related Information

The following commands: “**acct/***” on page 13, “**acctcms**” on page 18, “**acctcom**” on page 20, “**acctcon**” on page 24, “**acctmerg**” on page 28, “**acctprc**” on page 30, “**cron**” on page 220, and “**fwtmp**” on page 457.

The **acct** system call and the **acct** and **utmp** files in *AIX Operating System Technical Reference*.

"Running System Accounting" in *IBM RT Managing the AIX Operating System*.

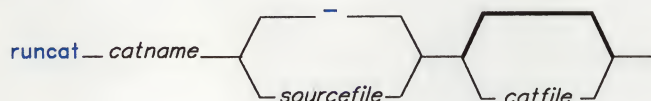
runcat

runcat

Purpose

Pipes data from **mkcatdefs** to **runcat**

Syntax



OL805491

Description

The **runcat** command invokes the **mkcatdefs** command and pipes the message source data (the output from **mkcatdefs**) to the **gencat** program.⁸ This method is simpler than using the redirection operator **>** to capture the output from **mkcatdefs** and then running **gencat**. The format for **runcat** is:

```
$ runcat catname sourcefile [catfile]
```

The file *sourcefile* contains the message text with your symbolic identifiers. **mkcatdefs** uses *catname* to generate the name of the symbolic definition file by adding **_msg.h** to the end of *catname*, and to generate the symbolic name for the catalog file by adding **MF_** to the beginning of *catname*. The definition file must be included in your application program. The symbolic name for the catalog file can be used in **catopen** or **NLcatopen** instead of the actual file name.

catfile is the name of the catalog file created by **gencat**. If you do not specify this parameter, **gencat** names the catalog file by adding **.cat** to the end of *catname*. This file name can also be used in **catopen** or **NLcatopen**.

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

⁸ **runcat** is an AIX extension to the X/Open standard.

Related Information

The following commands: “**dspcat**” on page 357, “**dspmsg**” on page 359, “**gencat**” on page 470, and “**mkcatdefs**” on page 651.

The **catopen**, **catgets**, **catgetamsg**, **catclose**, **NLcatopen**, **NLcatgets**, and **NLgetamsg** files in *AIX Operating System Technical Reference*.

The discussion of **runcat** in *AIX Operating System Programming Tools and Interfaces*.

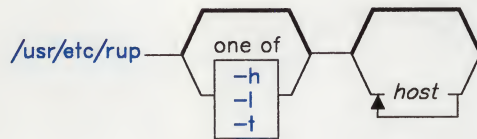
rup

rup

Purpose

Displays the status of hosts on a network.

Syntax



OL805492

Description

The **rup** command broadcasts a query on the local network and displays the responses it receives. It gives a status report on system usage times (uptimes) and load averages.

To query specific hosts, list their names as arguments following **rup**.

Note: Remote hosts respond only if running the **rstatd** daemon. The **rstatd** daemon is started from **inetd**.

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

Flags

When used without flags, **rup** displays the responses in the order it receives them. Use the following flags to change the display order.

- h** Displays responses alphabetically by host name.
- l** Displays responses by load average.
- t** Displays responses by up time.

Related Information

The following command: **"rstatd"** on page 847.

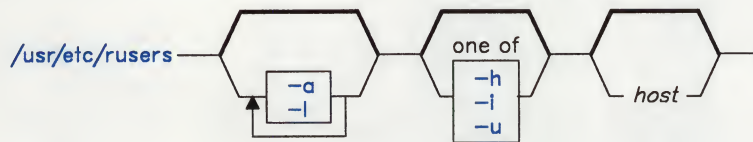
rusers

rusers

Purpose

Identifies users logged in on network hosts

Syntax



OL805496

Description

The **rusers** command broadcasts a query on the local network and displays the responses it receives.

To query specific hosts, list their names following the **rusers** command.

Note: Remote hosts will only respond if they are running the **rusersd** daemon (started from `/etc/inetd.conf`).

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

Flags

The **rusers** command displays the responses to the query in the order it receives them. Each machine is listed on a separate line. Use the following flags to change the print order:

- a** Responds for all machines even if no users are logged in.
- h** Sorts responses alphabetically by host name.
- i** Sorts responses by idle time. Idle time is reported if a user has not typed into the system for more than a minute.
- l** Responds with a longer listing in the style of the **who** command.

-u Sorts responses by numbers of users.

File

/etc/inetd.conf TCP/IP configuration file.

Related Information

The following commands: “**who**” on page 1219 and “**rusersd**” on page 858.

rusersd

rusersd

Purpose

Responds to queries from **rusers** command.

Syntax

`/usr/etc/rpc.rusersd`—|

OL805505

Description

The **rusersd** daemon allows remote hosts to respond to queries from the **rusers** command. The **inetd** daemon invokes **rusersd**.

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

File

`/etc/inetd.conf` TCP/IP configuration file.

Related Information

The following command: “**rusers**” on page 856.

rwall

Purpose

Writes to all users over a network.

Syntax

```
/usr/etc/rwall — -h host — -n netgroup — |
```

```
/usr/etc/rwall — -n netgroup — |
```

```
/usr/etc/rwall — host — |
```

OL805547

Description

The **rwall** command reads a message from standard input and broadcasts it to all users logged in to the specified host machines. It reads the message from standard input until it reaches an **end-of-file** character.

The **rwall** sends the message with the following introduction line:

Broadcast Message.....

Note: Users can only receive a message if they are running **rwalld**. This daemon is started by the **inetd** daemon.

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

Flags

- n Sends the message to specific network groups only. Network groups are defined in the **netgroup** file.

rwall

File

`/etc/netgroup`

Related Information

The following command: “**rwalld**” on page 861.

The **netgroup** file format in AIX Operating System Technical Reference.

rwalld

Purpose

Handles requests for the **rwall** and **shutdown** commands.

Syntax

`/usr/etc/rwalld` —|

OL805545

Description

The **rwalld** daemon handles requests from the **rwall** and **shutdown** commands. The **inetd** daemon invokes **rwalld**.

Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

File

`/etc/inetd.conf` TCP/IP configuration file.

Related Information

The following commands: “**rwall**” on page 859 and “**shutdown**” on page 946.


sact

sact

Purpose

Displays current Source Code Control System (SCCS) file editing status.

Syntax

sact 

OL805063

Description

The **sact.** command reads *Source Code Control System* (SCCS) files and writes to standard output the contents, if any, for the *p-file* associated with *file* (see “SCCS Files” on page 478 for information on the contents of the p-file). If - (minus) is specified for *file*, **sact.** reads standard input, and interprets each line as the name of an SCCS file. If *file* is a directory, **sact.** performs its actions on all SCCS files (that is, those files with the s. prefix).

Japanese Language Support Information

This command has not been modified to support Japanese characters.

Related Information

The following commands: “**delta**” on page 310, “**get**” on page 477, and “**unget**” on page 1116.

The **sccsfile** file in *AIX Operating System Technical Reference*.

The discussion of SCCS in *AIX Operating System Programming Tools and Interfaces*.